SIEMENS

SINUMERIK

SINUMERIK 840D sl / 828D
Preparação do trabalho

Manual de programação

Prefácio

Programação flexível de NC 1

Gerenciamento de arquivos e programas 2

Áreas de proteção 3

Comandos especiais de curso 4

Transformações de coordenadas (FRAMES) 5

Transformações 6

Corretores de ferramentas 7

Comportamento de percurso 8

Acoplamentos de eixo 9

Ações síncronas de movimentos 10

Oscilação 11

Estampagem e puncionamento 12

Retificação 13

Outras funções 14

Programas de desbaste próprios 15

Tabelas 16

Apêndice A

Válido para

Comando
SINUMERIK 840D sl / 840DE sl
SINUMERIK 828D

Software	Versão
NCU software de sistema	2.6 SP1

03/2010
6FC5398-2BP20-1KA0

Informações jurídicas

Conceito de aviso

Este manual contém instruções que devem ser observadas para sua própria segurança e também para evitar danos materiais. As instruções que servem para sua própria segurança são sinalizadas por um símbolo de alerta, as instruções que se referem apenas à danos materiais não são acompanhadas deste símbolo de alerta. Dependendo do nível de perigo, as advertências são apresentadas como segue, em ordem decrescente de gravidade.

⚠PERIGO
significa que haverá caso de morte ou lesões graves**,** caso as medidas de segurança correspondentes não forem tomadas.

⚠AVISO
significa que haverá caso de morte ou lesões graves**,** caso as medidas de segurança correspondentes não forem tomadas.

⚠CUIDADO
acompanhado do símbolo de alerta, indica um perigo iminente que pode resultar em lesões leves, caso as medidas de segurança correspondentes não forem tomadas.

CUIDADO
não acompanhado do símbolo de alerta, significa que podem ocorrer danos materiais, caso as medidas de segurança correspondentes não forem tomadas.

ATENÇÃO
significa que pode ocorrer um resultado ou um estado indesejados, caso a instrução correspondente não for observada.

Ao aparecerem vários níveis de perigo, sempre será utilizada a advertência de nível mais alto de gravidade. Quando é apresentada uma advertência acompanhada de um símbolo de alerta relativamente a danos pessoais, esta mesma também pode vir adicionada de uma advertência relativa a danos materiais.

Pessoal qualificado

O produto/sistema, ao qual esta documentação se refere, só pode ser manuseado por **pessoal qualificado** para a respectiva definição de tarefas e respeitando a documentação correspondente a esta definição de tarefas, em especial as indicações de segurança e avisos apresentados. Graças à sua formação e experiência, o pessoal qualificado é capaz de reconhecer os riscos do manuseamento destes produtos/sistemas e de evitar possíveis perigos.

Utilização dos produtos Siemens em conformidade com as especificações

Tenha atenção ao seguinte:

⚠AVISO
Os produtos da Siemens só podem ser utilizados para as aplicações especificadas no catálogo e na respetiva documentação técnica. Se forem utilizados produtos e componentes de outros fornecedores, estes têm de ser recomendados ou autorizados pela Siemens. Para garantir um funcionamento em segurança e correto dos produtos é essencial proceder corretamente ao transporte, armazenamento, posicionamento, instalação, montagem, colocação em funcionamento, operação e manutenção. Devem-se respeitar as condições ambiente autorizadas e observar as indicações nas respetivas documentações.

Marcas

Todas denominações marcadas pelo símbolo de propriedade autoral ® são marcas registradas da Siemens AG. As demais denominações nesta publicação podem ser marcas em que os direitos de proprietário podem ser violados, quando usadas em próprio benefício, por terceiros.

Exclusão de responsabilidade

Nós revisamos o conteúdo desta documentação quanto a sua coerência com o hardware e o software descritos. Mesmo assim ainda podem existir diferenças e nós não podemos garantir a total conformidade. As informações contidas neste documento são revisadas regularmente e as correções necessárias estarão presentes na próxima edição.

Siemens AG
Industry Sector
Postfach 48 48
90026 NÜRNBERG
ALEMANHA

N.º de encomenda de documento: 6FC5398-2BP20-1KA0
℗ 04/2010

Copyright © Siemens AG 2010.
Sujeito a modificações sem aviso prévio

Prefácio

Documentação SINUMERIK®

A documentação SINUMERIK está organizada em 3 categorias:

- Documentação geral
- Documentação do usuário
- Documentação do fabricante e assistência técnica

Através do link http://www.siemens.com/motioncontrol/docu encontra-se informações do seguinte tema:

- Ordering documentation
  Aqui encontra-se uma lista da documentação atual impressa.
- Download documentation
  Links adicionais para o download de arquivos de Service & Support.
- (Online) research in the documentation
  Informações do DOConCD e acesso direto aos documentos no DOConWEB.
- Documentação do conteúdo básico individual Siemens organizado com o My Documentation Manager (MDM), vide http://www.siemens.com/mdm

  O My Documentation Manager lhe oferece uma série de características para criar sua própria documentação de máquina.
- Treinamentos e FAQs

  As informações sobre o treinamento oferecido e sobre as FAQ's (frequently asked questions) estão disponíveis em:

Grupo destino

Esta publicação é dirigida a:

- Programadores
- Projetistas

Aplicação

O manual de programação possibilita a criação de progamas e interface de software para editar, testar e para corrigir erros.

Escopo padrão

Este manual de programação descreve as funcionalidades de escopo padrão. As complementações e alterações realizadas pelo fabricante da máquina são documentadas pelo fabricante da máquina.

No comando podem existir outras funções que não foram explicadas nesta documentação. Isso, no entanto, não implica nenhuma obrigação destas funções serem fornecidas com um novo controle ou em caso de serviço.

Da mesma forma, devido à grande variedade de itens, esta documentação não compreende todas as informações detalhadas de todos os tipos de produto, e também não podem ser considerados todos os casos possíveis de instalação, operação e manutenção.

Suporte técnico

Para dúvidas entre em contato com nosso Hotline:

	Europa / África
Telefone	+49 180 5050 - 222
Fax	+49 180 5050 - 223
0,14 €/Min. na rede fixa alemã, possíveis divergências para tarifas de celular	
Internet	http://www.siemens.com/automation/support-request

	América
Telefone	+1 423 262 2522
Fax	+1 423 262 2200
E-Mail	mailto:techsupport.sea@siemens.com

	Ásia / Pacífico
Telefone	+86 1064 757575
Fax	+86 1064 747474
E-Mail	mailto:support.asia.automation@siemens.com

Indicação

Os números de telefone para suporte técnico de cada país estão disponíveis na Internet: http://www.automation.siemens.com/partner

Perguntas sobre a documentação

Em caso de dúvidas sobre documentação (reclamações, correções) favor encaminhar Fax ou E-Mail ao seguinte endereço:

Fax:	+49 9131- 98 2176
E-Mail:	mailto:docu.motioncontrol@siemens.com

O modelo de fax se encontra no anexo do documento.

Endereço de Internet para SINUMERIK

http://www.siemens.com/sinumerik

Manual de programação "Fundamentos" e "Preparação de trabalho"

A descrição da programação de NC é dividida em 2 manuais:

1. **Fundamentos**

   O manual de programação básico é voltado para o operador de máquinas com conhecimentos específicos em fresamento, furação e torneamento. Exemplos simples de programação são usados para explicar as instruções, que também são definidas pela DIN 66025.

2. **Preparação do trabalho**

   O manual de programação "Preparação de trabalho" oferece ao técnico, conhecimentos sobre todas as possibilidades de programação. O Comando SINUMERIK permite que com uma linguagem de programação especial sejam feitos complexos programas de peça (por exemplo, superfícies de formas livres, sincronismo de canais, ...) e facilita a programação de operações de alta complexidade.

Disponibilidade dos elementos da linguagem de NC descritos

Todos o elementos de linguagem de NC descritos no seguinte manual são disponíveis para SINUMERIK 840D sl. A disponibilidade com relação ao SINUMERIK 828D está indicada na coluna "828D" em "Lista de instruções (Página 731)".

Conteúdo

Prefácio 3

1 Programação flexível de NC 17

1.1 Variáveis 17
1.1.1 Informações gerais sobre variáveis 17
1.1.2 Variáveis de sistema 18
1.1.3 Variáveis de usuário pré-definidas: Parâmetros de cálculo (R) 20
1.1.4 Variáveis de usuário pré-definidas: Variáveis de ligação 22
1.1.5 Definição de variáveis de usuário (DEF) 24
1.1.6 Redefinição de variáveis de sistema, variáveis de usuário e comandos de linguagem NC (REDEF) 30
1.1.7 Atributo: Valor de inicialização 33
1.1.8 Atributo: Valores de limite (LLI, ULI) 37
1.1.9 Atributo: Unidade física (PHU) 38
1.1.10 Atributo: Direitos de acesso (APR, APW, APRP, APWP, APRB, APWB) 41
1.1.11 Visão geral dos atributos definíveis e redefiníveis 46
1.1.12 Definição e inicialização de variáveis de campo (DEF, SET, REP) 47
1.1.13 Definição e inicialização de variáveis de campo (DEF, SET, REP): Outras informações 52
1.1.14 Tipos de dados 55

1.2 Programação indireta 56
1.2.1 Programação indireta de endereços 56
1.2.2 Programação indireta de códigos G 59
1.2.3 Programação indireta de atributos de posição (GP) 60
1.2.4 Programação indireta de linhas de programa de peça (EXECSTRING) 63

1.3 Funções de cálculo 64

1.4 Operadores de comparação e operadores lógicos 67

1.5 Correção da precisão em erros de comparação (TRUNC) 69

1.6 Mínimo, máximo e área de variáveis (MINVAL, MAXVAL, BOUND) 71

1.7 Prioridade das operações 73

1.8 Possíveis conversões de tipo 74

1.9 Operações de String 75
1.9.1 Conversão de tipos para STRING (AXSTRING) 76
1.9.2 Conversão de tipos de STRING (NUMBER, ISNUMBER, AXNAME) 77
1.9.3 Encadeamento de Strings (<<) 78
1.9.4 Mudança para letras minúsculas / letras maiúsculas (TOLOWER, TOUPPER) 80
1.9.5 Definir o tamanho de uma Strings (STRLEN) 81
1.9.6 Localizar caractere/String na String (INDEX, RINDEX, MINDEX, MATCH) 81
1.9.7 Seleção de uma String parcial (SUBSTR) 83
1.9.8 Seleção de um caractere individual (STRINGVAR, STRINGFELD) 83

1.10 Saltos e ramificações de programa 85
1.10.1 Salto de retorno ao início do programa (GOTOS) 85
1.10.2 Saltos de programa até marcadores de salto (GOTOB, GOTOF, GOTO, GOTOC) 86
1.10.3 Bifurcação do programa (CASE ... OF ... DEFAULT ...) 89

1.11 Repetição de partes do programa (REPEAT, REPEATB, ENDLABEL, P) 91

1.12 Estruturas de controle ... 98
1.12.1 Loop de programa com alternativa (IF, ELSE, ENDIF) ... 99
1.12.2 Loop de programa infinito (LOOP, ENDLOOP) ... 101
1.12.3 Loop contador (FOR ... TO ..., ENDFOR) ... 102
1.12.4 Loop de programa com condição no início do loop (WHILE, ENDWHILE) ... 103
1.12.5 Loop de programa com condição no fim do loop (REPEAT, UNTIL) ... 104
1.12.6 Exemplo de programa com estruturas de controle aninhadas ... 105
1.13 Coordenação de programa (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM) ... 106
1.14 Rotina de interrupção (ASUP) ... 112
1.14.1 Funcionamento de uma rotina de interrupção ... 112
1.14.2 Criar rotina de interrupção ... 113
1.14.3 Atribuição e partida de rotinas de interrupção (SETINT, PRIO, BLSYNC) ... 114
1.14.4 Desativar / ativar a atribuição de uma rotina de interrupção (DISABLE, ENABLE) ... 116
1.14.5 Deletar a atribuição de uma rotina de interrupção (CLRINT) ... 117
1.14.6 Retração rápida do contorno (SETINT LIFTFAST, ALF) ... 118
1.14.7 Sentido de deslocamento na retração rápida do contorno ... 120
1.14.8 Execução de movimentos com rotinas de interrupção ... 123
1.15 Troca de eixos, troca de fusos (RELEASE, GET, GETD) ... 124
1.16 Transferir eixo de outro canal (AXTOCHAN) ... 129
1.17 Ativar dados de máquina (NEWCONF) ... 131
1.18 Gravar arquivo (WRITE) ... 132
1.19 Deletar arquivo (DELETE) ... 135
1.20 Ler linhas no arquivo (READ) ... 137
1.21 Controle de presença de um arquivo (ISFILE) ... 141
1.22 Leitura das informações de arquivo (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO) ... 143
1.23 Cálculo de checksum através de um campo (CHECKSUM) ... 146
1.24 Arredondamento (ROUNDUP) ... 148
1.25 Técnica de subrotinas ... 149
1.25.1 Informações gerais ... 149
1.25.1.1 Subrotina ... 149
1.25.1.2 Nomes de subrotina ... 150
1.25.1.3 Aninhamento de subrotinas ... 151
1.25.1.4 Caminho de localização ... 153
1.25.1.5 Parâmetros formais e parâmetros atuais ... 153
1.25.1.6 Transferência de parâmetros ... 154
1.25.2 Definição de uma subrotina ... 156
1.25.2.1 Subrotina sem transferência de parâmetros ... 156
1.25.2.2 Subrotina com transferência de parâmetros Call-by-Value (PROC) ... 157
1.25.2.3 Subrotina com transferência de parâmetros Call-by-Reference (PROC, VAR) ... 159
1.25.2.4 Salvar funções G modais (SAVE) ... 161
1.25.2.5 Supressão de processamento bloco a bloco (SBLOF, SBLON) ... 162
1.25.2.6 Supressão da atual exibição de bloco (DISPLOF, DISPLON, ACTBLOCNO) ... 168
1.25.2.7 Identificar subrotinas com preparação (PREPRO) ... 172
1.25.2.8 Salto de retorno para a subrotina M17 ... 173
1.25.2.9 Salto de retorno para subrotina RET ... 174
1.25.2.10 Salto de retorno parametrizável da subrotina (RET ...) ... 175
1.25.3 Chamada de uma subrotina ... 182

1.25.3.1 Chamada de subrotina sem transferência de parâmetros 182
1.25.3.2 Chamada de subrotina com transferência de parâmetros (EXTERN) 184
1.25.3.3 Número de repetições de programa (P) 187
1.25.3.4 Chamada modal de subrotina (MCALL) 189
1.25.3.5 Chamada de subrotinas indireta (CALL) 191
1.25.3.6 Chamada de subrotina indireta com indicação da parte de programa que deve ser executada (CALL BLOCK ... TO ...) 192
1.25.3.7 Chamada indireta de um programa programado em linguagem ISO (ISOCALL) 194
1.25.3.8 Chamada de subrotina com indicação de caminho e parâmetros (PCALL) 195
1.25.3.9 Ampliar o caminho de localização na chamada de subrotinas (CALLPATH) 196
1.25.3.10 Executar subrotina externa (EXTCALL) 197
1.25.4 Ciclos 201
1.25.4.1 Ciclos: Parametrizar ciclos de usuário 201
1.26 Técnica de macros (DEFINE ... AS) 205

2 Gerenciamento de arquivos e programas 209

2.1 Memória de programa 209
2.2 Memória de trabalho (CHANDATA, COMPLETE, INITIAL) 214
2.3 Instrução de estruturação no editor Step (SEFORM) 217

3 Áreas de proteção 219

3.1 Definição das áreas de proteção (CPROTDEF, NPROTDEF) 219
3.2 Ativar/desativar áreas de proteção (CPROT, NPROT) 223
3.3 Verificação quanto à violação da área de proteção, limite da área de trabalho e limites de software (CALCPOSI) 227

4 Comandos especiais de curso 235

4.1 Aproximar posições codificadas (CAC, CIC, CDC, CACP, CACN) 235
4.2 Interpolação de Spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) 237
4.3 Agrupamento de Spline (SPLINEPATH) 249
4.4 Compressão de blocos NC (COMPON, COMPCURV, COMPCAD, COMPOF) 251
4.5 Interpolação de polinômios (POLY, POLYPATH, PO, PL) 254
4.6 Referência ajustável do percurso (SPATH, UPATH) 260
4.7 Medição com apalpador comutável (MEAS, MEAW) 263
4.8 Função de medição ampliada (MEASA, MEAWA, MEAC) (opcional) 266
4.9 Funções especiais para usuários OEM (OEMIPO1, OEMIPO2, G810 até G829) 276
4.10 Redução de avanço com desaceleração nos cantos (FENDNORM, G62, G621) 277
4.11 Critérios programáveis para fim de movimentos (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA) 278
4.12 Bloco programável de parâmetros servo (SCPARA) 282

5 Transformações de coordenadas (FRAMES) 283

5.1 Transformação de coordenadas através de variável Frame 283
5.1.1 Variável Frame pré-definida ($P_BFRAME, $P_IFRAME, $P_PFRAME, $P_ACTFRAME) 285
5.2 Atribuir valores para variáveis Frame / Frame 291

5.2.1 Atribuir valores diretos (valor de eixo, ângulo, escala) ... 291
5.2.2 Ler e alterar componentes de Frame (TR, FI, RT, SC, MI) ... 294
5.2.3 Referências de Frames completos ... 295
5.2.4 Definição de novos Frames (DEF FRAME) ... 297
5.3 Deslocamento aproximado e deslocamento fino (CFINE, CTRANS) ... 298
5.4 Deslocamento externo de ponto zero ... 300
5.5 Deslocamento Preset (PRESETON) ... 301
5.6 Cálculo de Frame a partir de 3 pontos de medição no espaço (MEAFRAME) ... 303
5.7 Frames globais da NCU ... 307
5.7.1 Frames específicos de canal ($P_CHBFR, $P_UBFR) ... 308
5.7.2 Frames ativos no canal ... 309
6 Transformações ... 315
6.1 Programação geral de tipos de transformação ... 315
6.1.1 Movimentos de orientação nas transformações ... 318
6.1.2 Vista geral da transformação de orientação TRAORI ... 321
6.2 Transformação de três, quatro e cinco eixos (TRAORI) ... 323
6.2.1 Relações gerais do cabeçote de ferramenta cardânico ... 323
6.2.2 Transformação de três, quatro e cinco eixos (TRAORI) ... 326
6.2.3 Variantes da programação da orientação e posição básica (ORIRESET) ... 327
6.2.4 Programação da orientação da ferramenta (A..., B..., C..., LEAD, TILT) ... 329
6.2.5 Fresamento de topo (fresamento 3D A4, B4, C4, A5, B5, C5) ... 335
6.2.6 Relação dos eixos de orientação (ORIWKS, ORIMKS) ... 337
6.2.7 Programação dos eixos de orientação (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2) ... 339
6.2.8 Programação da orientação ao longo de uma superfície periférica cônica (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO) ... 342
6.2.9 Especificação de orientação de dois pontos de contato (ORICURVE, PO[XH]=, PO[YH]=, PO[ZH]=) ... 346
6.3 Polinômios de orientação (PO[ângulo], PO[coordenadas]) ... 348
6.4 Rotações da orientação da ferramenta (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) ... 350
6.5 Orientações relativas à trajetória ... 353
6.5.1 Tipos de orientação relativos à trajetória ... 353
6.5.2 Rotação da orientação da ferramenta relativa à trajetória (ORIPATH, ORIPATHS, ângulo de rotação) ... 355
6.5.3 Interpolação relativa à trajetória da rotação da ferramenta (ORIROTC, THETA) ... 357
6.5.4 Suavização do decurso de orientação (ORIPATHS A8=, B8=, C8=) ... 359
6.6 Compressão da orientação (COMPON, COMPCURV, COMPCAD) ... 361
6.7 Suavização do decurso de orientação (ORISON, ORISOF) ... 365
6.8 Transformação cinemática ... 367
6.8.1 Fresamento em peças torneadas (TRANSMIT) ... 367
6.8.2 Transformação de superfície cilíndrica (TRACYL) ... 371
6.8.3 Eixo inclinado (TRAANG) ... 380
6.8.4 Programar eixo inclinado (G05, G07) ... 383
6.9 Deslocamento PTP cartesiano ... 385
6.9.1 PTP com TRANSMIT ... 390
6.10 Condições gerais na seleção de uma transformação ... 394

6.11 Desselecionar a transformação (TRAFOOF) 395
6.12 Transformações encadeadas (TRACON, TRAFOOF) 396
7 Corretores de ferramentas 399
7.1 Memória de correções 399
7.2 Corretores aditivos 403
7.2.1 Ativar correções aditivas (DL) 403
7.2.2 Definir valores de desgaste e de ajuste ($TC_SCPxy[t,d], $TC_ECPxy[t,d]) 405
7.2.3 Deletar correções aditivas (DELDL) 406
7.3 Tratamentos especiais de corretor de ferramenta 407
7.3.1 Espelhar comprimentos de ferramentas 409
7.3.2 Avaliação de sinais de desgaste 410
7.3.3 Sistema de coordenadas da usinagem ativa (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS) 411
7.3.4 Comprimento de ferramenta e mudança de planos 414
7.4 Correção de ferramenta Online (PUTFTOCF, FCTDEF, PUTFTOC, FTOCON, FTOCOF) 415
7.5 Ativação da correção de ferramenta 3D (CUT3DC..., CUT3DF...) 420
7.5.1 Ativação das correções de ferramentas 3D (CUT3DC, CUT3DF, CUT3DFS, CUT3DFF, ISD) 420
7.5.2 Correções de ferramenta 3D: Fresamento periférico, fresamento de topo 422
7.5.3 Correções de ferramenta 3D: Formas e dados de ferramenta para fresamento de topo 424
7.5.4 Correções de ferramenta 3D: Correção na trajetória, curvatura de trajetória, profundidade de imersão (CUT3DC, ISD) 426
7.5.5 Correções de ferramenta 3D: Cantos internos/externos e procedimento de ponto de intersecção (G450/G451) 428
7.5.6 Correções de ferramenta 3D: Fresamento periférico 3D com superfícies de limitação 430
7.5.7 Correções de ferramenta 3D: Consideração de uma superfície de limitação (CUT3DCC, CUT3DCCD) 430
7.6 Orientação da ferramenta (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST) 434
7.7 Seleção livre de número D, número de gume 440
7.7.1 Atribuição livre de números D, número de corte (endereço CE) 440
7.7.2 Atribuição livre de números D: Verificar números D (CHKDNO) 441
7.7.3 Atribuição livre de números D: Renomear números D (GETDNO, SETDNO) 442
7.7.4 Atribuição livre de números D: Determinar o número T para o número D especificado (GETACTTD) 443
7.7.5 Atribuição livre de números D: Invalidar números D (DZERO) 444
7.8 Cinemática do porta-ferramenta 445
7.9 Correção de comprimento de ferramenta para porta-ferramentas orientáveis (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ) 451
7.10 Correção Online de comprimento de ferramenta (TOFFON, TOFFOF) 455
7.11 Modificação dos dados de corte para ferramentas orientáveis (CUTMOD) 458
8 Comportamento de percurso 465
8.1 Controle tangencial (TANG, TANGON, TANGOF, TLIFT, TANGDEL) 465
8.2 Característica de avanço (FNORM, FLIN, FCUB, FPO) 472
8.3 Execução do programa com memória de pré-processamento (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE) 477

8.4 Segmentos de programa interrompidos condicionalmente (DELAYFSTON, DELAYFSTOF) ... 480
8.5 Evitar posição de programa para SERUPRO (IPTRLOCK, IPTRUNLOCK) ... 485
8.6 Reaproximação no contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN) ... 488
8.7 Controle na condução do movimento ... 498
8.7.1 Correção percentual de solavancos (JERKLIM) ... 498
8.7.2 Correção percentual da velocidade (VELOLIM) ... 499
8.7.3 Exemplo de programa para JERKLIM e VELOLIM ... 502
8.8 Tolerância de contorno/orientação programável (CTOL, OTOL, ATOL) ... 503
8.9 Tolerância em movimentos G0 (STOLF) ... 507

9 Acoplamentos de eixo ... 509

9.1 Movimento acoplado (TRAILON, TRAILOF) ... 509
9.2 Tabelas de curvas (CTAB) ... 513
9.2.1 Definição de tabelas de curvas (CTABDEF, CATBEND) ... 514
9.2.2 Controle de presença de uma tabela de curvas (CTABEXISTS) ... 520
9.2.3 Eliminação de tabelas de curvas (CTABDEL) ... 520
9.2.4 Bloqueio de tabelas de curvas contra eliminação e sobregravação (CTABLOCK, CTABUNLOCK) ... 522
9.2.5 Tabelas de curvas: Determinação de propriedades de tabela (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD) ... 523
9.2.6 Leitura dos valores da tabela de curvas (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX) ... 525
9.2.7 Tabelas de curvas: Controle do aproveitamento de recursos (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL) ... 530
9.3 Acoplamento axial de valor mestre (LEADON, LEADOF) ... 532
9.4 Caixa de transmissão eletrônica (EG) ... 538
9.4.1 Definir caixa de transmissão eletrônica (EGDEF) ... 538
9.4.2 Ativar caixa de transmissão eletrônica (EGON, EGONSYN, EGONSYNE) ... 540
9.4.3 Desativar transmissão eletrônica (EGOFS, EGOFC) ... 543
9.4.4 Deletar a definição de uma caixa de transmissão eletrônica (EGDEL) ... 544
9.4.5 Avanço por rotação (G95) / caixa de transmissão eletrônica (FPR) ... 545
9.5 Fuso sincronizado ... 546
9.5.1 Fuso sincronizado: Programação (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC) ... 547
9.6 Grupo mestre/escravo (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS) ... 558

10 Ações síncronas de movimentos ... 563

10.1 Fundamentos ... 563
10.1.1 Área de validade e seqüência de usinagem (ID, IDS) ... 565
10.1.2 Controle cíclico da condição (WHEN, WHENEVER, FROM, EVERY) ... 567
10.1.3 Ações (DO) ... 569
10.2 Operadores para condições e ações ... 570
10.3 Variáveis de processamento principal para ações síncronas ... 572
10.3.1 Variáveis de sistema ... 572
10.3.2 Conversão de tipo implícita ... 574
10.3.3 Variáveis GUD ... 575

10.3.4 Identificadores de eixo default (NO_AXIS) 577
10.3.5 Marcador de ação sincronizada ($AC_MARKER[n]) 578
10.3.6 Parâmetro de ação sincronizada ($AC_PARAM[n]) 579
10.3.7 Parâmetros de cálculo ($R[n]) 580
10.3.8 Ler e gravar dados de máquina NC e dados de ajuste NC 581
10.3.9 Variável de temporizador ($AC_TIMER[n]) 582
10.3.10 Variáveis FIFO ($AC_FIFO1[n] ... $AC_FIFO10[n]) 583
10.3.11 Informação sobre os tipos de bloco no interpolador ($AC_BLOCKTYPE, $AC_BLOCKTYPEINFO, $AC_SPLITBLOCK) 585

10.4 Ações em ações síncronas 588
10.4.1 Vista geral das possíveis ações em ações sincronizadas 588
10.4.2 Emissão de funções auxiliares 591
10.4.3 Definir bloqueio de leitura (RDISABLE) 592
10.4.4 Cancelar a parada de pré-processamento (STOPREOF) 593
10.4.5 Anular curso restante (DELDTG) 594
10.4.6 Definição de polinômio (FCTDEF) 596
10.4.7 Função sincronizada (SYNFCT) 599
10.4.8 Controle de distância com correção limitada ($AA_OFF_MODE) 602
10.4.9 Correção Online da ferramenta (FTOC) 605
10.4.10 Correção Online de comprimento de ferramenta ($AA_TOFF) 608
10.4.11 Movimentos de posicionamento 610
10.4.12 Posicionar eixo (POS) 611
10.4.13 Posição na área de referência especificada (POSRANGE) 613
10.4.14 Iniciar/parar eixo (MOV) 614
10.4.15 Troca de eixos (RELEASE, GET) 615
10.4.16 Avanço por eixo (FA) 619
10.4.17 Chave fim de curso de SW 619
10.4.18 Coordenação de eixo 620
10.4.19 Definir valor real (PRESETON) 621
10.4.20 Movimentos de fuso 622
10.4.21 Movimento acoplado (TRAILON, TRAILOF) 623
10.4.22 Acoplamento de valor mestre (LEADON, LEADOF) 625
10.4.23 Medição (MEAWA, MEAC) 628
10.4.24 Inicialização de variáveis de campo (SET, REP) 629
10.4.25 Definir/deletar marcadores de espera (SETM, CLEARM) 630
10.4.26 Reações à erros (SETAL) 631
10.4.27 Deslocar até o encosto fixo (FXS, FXST, FXSW, FOCON, FOCOF) 632
10.4.28 Definição do ângulo de tangente de percurso em ações sincronizadas 635
10.4.29 Determinação do atual Override 635
10.4.30 Avaliação do tempo usado das ações sincronizadas 636

10.5 Ciclos tecnológicos 638
10.5.1 Variável de contexto ($P_TECCYCLE) 641
10.5.2 Parâmetro Call-by-Value 642
10.5.3 Inicialização de parâmetros Default 642
10.5.4 Controle da execução de ciclos de tecnologia (ICYCOF, ICYCON) 643
10.5.5 Concatenação de ciclos de tecnologia 644
10.5.6 Ciclos de tecnologia em ações sincronizadas por blocos 644
10.5.7 Estruturas de controle (IF) 644
10.5.8 Instruções de salto (GOTO, GOTOF, GOTOB) 645
10.5.9 Bloquear, liberar, resetar (LOCK, UNLOCK, RESET) 646

10.6 Cancelar ação sincronizada (CANCEL) 648

10.7 Comportamento de comando em determinados estados operacionais 649

11 Oscilação 653
11.1 Oscilação assíncrona (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB) 653
11.2 Oscilação controlada através de ações síncronas (OSCILL) 659
12 Estampagem e puncionamento 667
12.1 Ativação, desativação 667
12.1.1 Estampagem e puncionamento ativado ou desativado (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC) 667
12.2 Preparação automática do curso 672
12.2.1 Segmentação de curso para eixos de percurso 675
12.2.2 Segmentação de curso para eixos individuais 677
13 Retificação 679
13.1 Monitoração de ferramenta específica de retificação no programa de peça (TMON, TMOF) 679
14 Outras funções 681
14.1 Funções de eixo (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) 681
14.2 Eixos geométricos comutáveis (GEOAX) 684
14.3 Contentor de eixo (AXCTSWE, AXCTSWED) 689
14.4 Espera pela posição de eixo válida (WAITENC) 693
14.5 Controlar o escopo disponível de linguagem NC (STRINGIS) 695
14.6 Chamada de função ISVAR e leitura do índice Array de dados de máquina 699
14.7 Adaptação de curvas características de compensação (QECLRNON, QECLRNOF) 701
14.8 Chamada interativa a partir do programa de peça (MMC) 703
14.9 Tempo de processamento do programa / contador de peças 704
14.9.1 Tempo de processamento do programa / contador de peças (resumo) 704
14.9.2 Tempo de processamento do programa 704
14.9.3 Contador de peças 709
14.10 Alarmes (SETAL) 711
15 Programas de desbaste próprios 713
15.1 Funções de apoio para o desbaste 713
15.2 Criar tabela de contorno (CONTPRON) 714
15.3 Criar tabela de contorno codificada (CONTDCON) 720
15.4 Determinar a intersecção entre dois elementos de contorno (INTERSEC). 724
15.5 Executar por blocos os elementos de contorno de uma tabela (EXECTAB) 726
15.6 Calcular dados de círculos (CALCDAT) 727
15.7 Desativar a preparação de contorno (EXECUTE) 729
16 Tabelas 731
16.1 Lista de instruções 731

A Apêndice 805
A.1 Lista de abreviações 805
A.2 Feedback sobre a documentação 810
A.3 Vista Geral da documentação 812
Glossário 813
Índice 835

Programação flexível de NC 1

1.1 Variáveis

1.1.1 Informações gerais sobre variáveis

Através do uso de variáveis, principalmente junto com funções de cálculo e estruturas de controle, os programas de peça e os ciclos podem ser compostos de modo extremamente flexível. Para isso o sistema disponibiliza três diferentes tipos de variáveis:

- Variáveis de sistema

  As variáveis de sistema são variáveis definidas no sistema e disponibilizadas ao usuário, e possuem um significado fixo. Elas também são lidas e gravadas pelo software do sistema. Exemplo: Dados de máquina

  O significado de uma variável de sistema é amplamente definido de modo fixo pelo sistema através das propriedades. Porém, apenas uma pequena parte das propriedades permite ser adaptada pelo usuário através de uma redefinição. Veja ""

- Variáveis de usuário

  As variáveis de usuário são variáveis, cujo significado não é conhecido pelo sistema e nem pode ser avaliado pelo sistema. O significado é definido exclusivamente pelo usuário.

  As variáveis de usuário estão divididas em:

  – Variáveis de usuário pré-definidas

    As variáveis de usuário pré-definidas são variáveis já definidas no sistema, cuja grandeza ainda precisa ser parametrizada pelo usuário através de dados de máquina específicos. As propriedades destas variáveis podem ser amplamente adaptadas pelo usuário. Veja "".

  – Variáveis definidas pelo usuário

    As variáveis definidas pelo usuário são variáveis que são definidas exclusivamente pelo usuário e somente são criadas pelo sistema durante o processamento. Sua grandeza, tipo de dado, visibilidade e todas demais propriedades são definidas exclusivamente pelo usuário.

    Veja ""

1.1.2 Variáveis de sistema

As variáveis de sistema são variáveis pré-definidas no sistema, que em programas de peça e em ciclos oferecem o acesso à atual parametrização do comando, assim como aos estados da máquina, do comando e do processo.

Variáveis de pré-processamento

As variáveis de pré-processamento são denominadas como variáveis de sistema, que no contexto do pré-processamento, ou seja, no momento da interpretação do bloco do programa de peça onde está programada a variável de sistema, podem ser lidas e gravadas. As variáveis de pré-processamento não disparam nenhuma parada de pré-processamento.

Variáveis de processamento principal

As variáveis de processamento principal são denominadas como variáveis de sistema, que no contexto do processamento principal, ou seja, no momento da execução do bloco do programa de peça onde está programada a variável de sistema, podem ser lidas e gravadas. As variáveis de processamento principal são:

- Variáveis de sistema, que podem ser programadas em ações síncronas (leitura/gravação)
- Variáveis de sistema, que podem ser programadas no programa de peça e que disparam a parada de pré-processamento (leitura/gravação)
- Variáveis de sistema, que podem ser programadas no programa de peça e que determinam o valor no pré-processamento, mas a gravação somente ocorre no processamento principal (Sincronizado com o processamento principal: apenas gravação)

Sistemática de prefixo

Para uma identificação especial das variáveis de sistema normalmente se coloca um prefixo no nome, que é composto pelo caractere $, seguido de uma ou duas letras e um sublinhado:

$ + 1ª letra	Significado: Tipo de dados
Variáveis de sistema, que são lidas / gravadas no pré-processamento	
$M	Dados de máquina 1)
$S	Dados de ajuste, áreas de proteção 1)
$T	Dados de gerenciamento de ferramentas
$P	Valores programados
$C	Variáveis de ciclo dos ciclos envoltórios ISO
$O	Dados opcionais
R	Parâmetro R (parâmetro de cálculo) 2)
Variáveis de sistema, que são lidas / gravadas no processamento principal	
$$M	Dados de máquina 1)
$$S	Dados de ajuste 1)
$A	Atuais dados de processamento principal

$ + 1ª letra	Significado: Tipo de dados
$V	Dados servo
$R	Parâmetro R (parâmetro de cálculo) 2)
1) Para o uso de dados de máquina e dados de ajuste em programa de peça / ciclo como variáves de pré-processamento escreve-se o prefixo com um caractere $. Para o uso em ações síncronas como variável de processamento principal o prefixo é escrito com dois caracteres $. 2) Para o uso de um parâmetro R no programa de peça / ciclo como variável de pré-processamento não é escrito nenhum prefixo, p. ex. R10. Para o uso em uma ação síncrona como variável de processamento principal o prefixo é escrito com um caractere $, p. ex. $R10.	

2ª letra	Significado: Visibilidade
N	Variável global do NCK (**N**CK)
C	Variável específica de canal (**C**hannel)
A	Variável específica de eixo (**A**xis)

Condições gerais

Exceções na sistemática de prefixo

As seguintes variáveis de sistema diferem da sistemática de prefixo mencionada acima:

- $TC_...: A 2ª letra C neste caso não é uma variável específica de canal, mas variável de sistema específica de porta-ferramenta (TC = Tool Carrier)
- $P_ ...: Variáveis de sistema específicas de canal

Uso de dados de máquina e de dados de ajuste em ações síncronas

Para o uso de dados de máquina e de dados de ajuste em ações síncronas pode-se definir através do prefixo, se o dado de máquina ou de ajuste deve ser lido/gravado síncrono com o pré-processamento ou com o processamento principal.

Se o dado permanece inalterado durante o processamento, a leitura pode ser sincronizada com o pré-processamento. Para isso o prefixo do dado de máquina ou de ajuste é escrito com um caractere $:

Código de programa

```
ID=1 WHENEVER G710 $AA_IM[z] < $SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
```

Se o dado variar durante o processamento, a leitura / gravação deve ser sincronizada com o processamento principal. Para isso o prefixo do dado de máquina ou de ajuste é escrito com dois caracteres $:

Código de programa

```
ID=1 WHENEVER $AA_IM[z] < $$SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
```

Indicação

Gravação de dados de máquina

Na gravação de um dado de máquina ou de ajuste deve-se prestar atenção para que o nível de acesso ativo durante a execução do programa de peça / ciclo permita o acesso de gravação e que o efeito do dado seja "IMMEDIATE".

Literatura

Uma listagem das propriedades de todas as variáveis de sistema está disponível no(a):

/PGA1/ Manual de listas de variáveis de sistema

1.1.3 Variáveis de usuário pré-definidas: Parâmetros de cálculo (R)

Função

Os parâmetros de cálculo ou parâmetros R são variáveis de usuário pré-definidas com a identificação R, definida como campo do tipo de dado `REAL`. Por motivos históricos em parâmetros R é permitido usar tanto a forma escrita com índice de campo, p. ex. `R[10]`, como a forma escrita sem o índice de campo , p. ex. `R10`.

Para o uso em ações síncronas deve ser colocada a letra $ na frente, p. ex. `$R10`.

Sintaxe

Para uso como variável de pré-processamento:

```
R<n>
R[<expressão>]
```

Para uso como variável de processamento principal:

```
$R<n>
$R[<expressão>]
```

Significado

`R`: Identificador para uso como variável de pré-processamento, p. ex. em programa de peça

`$R`: Identificador para uso como variável de processamento principal, p. ex. em ações síncronas

Tipo: REAL

Faixa de valores: Para forma escrita não exponencial:

± (0.000 0001 ... 9999 9999)

Nota:
São permitidas no máximo 8 casas decimais

Para forma escrita exponencial:

± ($1*10^{-300}$ ... $1*10^{+300}$)

Nota:

- Forma escrita: <mantissa>EX<exponente> p. ex. 8.2EX-3
- São permitidos até 10 caracteres, inclusive o sinal precedente e o ponto decimal.

`<n>`: Número do parâmetro R

Tipo: INT

Faixa de valores: 0 - MAX_INDEX

Nota
O MAX_INDEX resulta do número parametrizado de parâmetros R:
MAX_INDEX = (MD28050 $MN_MM_NUM_R_PARAM) - 1

`<expressão>`: Índice de campo

Como índice de campo pode ser especificada uma expressão qualquer, enquanto o resultado da expressão permitir sua conversão para o tipo de dado `INT` (`INT`, `REAL`, `BOOL`, `CHAR`)

Exemplo

Atribuições em parâmetros R e uso de parâmetros R em funções matemáticas:

```
Código de programa          Comentário
R0=3.5678                   ;   Atribuição no pré-processamento
R[1]=-37.3                  ;   Atribuição no pré-processamento
R3=-7                       ;   Atribuição no pré-processamento
$R4=-0.1EX-5                ;   Atribuição no processamento principal: R4 = -0.1 *
                                10^-5
$R[6]=1.874EX8              ;   Atribuição no processamento principal: R6 = 1.874 *
                                10^8
R7=SIN(25.3)                ;   Atribuição no pré-processamento
```

Código de programa	Comentário
`R[R2]=R10`	`;` Endereçamento indireto através de parâmetro R
`R[(R1+R2)*R3]=5`	`;` Endereçamento indireto através de expressão matemática
`X=(R1+R2)`	`;` Desloca o eixo X até a posição resultante da soma de R1 e R2
`Z=SQRT(R1*R1+R2*R2)`	`;` Desloca o eixo Z até a posição da raiz quadrada de (R1^2 + R2^2)

1.1.4 Variáveis de usuário pré-definidas: Variáveis de ligação

Função

Através das variáveis de ligação durante a função "NCU-Link" podem ser trocados dados cíclicos entre NCUs, que estão ligadas uma a outra através de uma rede. Aqui elas permitem o acesso específico de formato de dado à memória de variáveis de ligação. A memória de variáveis de ligação é definida de acordo com a instalação, tanto pelo tamanho como pela estrutura de dados do usuário / fabricante da máquina.

Variáveis de ligação são variáveis de usuário globais de sistema, que podem ser lidas e gravadas em programas de peça e em ciclos mediante uma comunicação de ligação (Link) configurada de todas NCUs do grupo de ligação. Em contrapartida às variáveis de usuário globais (GUD) as variáveis de ligação também podem ser utilizadas em ações síncronas.

Para instalações sem NCU-Link ativo, as variáveis de ligação locais do comando podem ser utilizadas paralelamente às variáveis de usuário globais (GUD) como variáveis de usuário globais extras.

Sintaxe

```
$A_DLB[<índice>]
$A_DLW[<índice>]
$A_DLD[<índice>]
$A_DLR[<índice>]
```

Significado

`$A_DLB`: Variável de ligação para formato de dado BYTE (1 Byte)
- Tipo de dado: UINT
- Faixa de valores: 0 ... 255

`$A_DLW`: Variável de ligação para formato de dado WORD (2 Bytes)
- Tipo de dado: INT
- Faixa de valores: -32768 ... 32767

`$A_DLD`: Variável de ligação para formato de dado DWORD (4 Bytes)
- Tipo de dado: INT
- Faixa de valores: -2147483648 ... 2147483647

`$A_DLR`: Variável de ligação para formato de dado REAL (8 Bytes)

Tipo de dado: REAL

Faixa de valores: $\pm(2{,}2*10^{-308} \ldots 1{,}8*10^{+308})$

`<indice>`: Índice de endereço em Byte, calculado do início da memória de variáveis de ligação

Tipo de dado: INT

Faixa de valores: 0 - MAX_INDEX

Nota

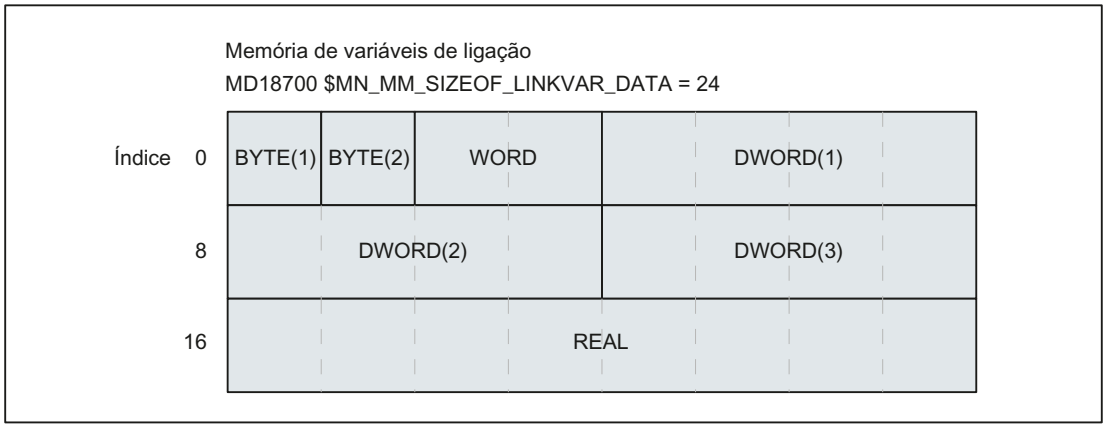
- O MAX_INDEX resulta do tamanho parametrizado da memória de variáveis de ligação: MAX_INDEX = (MD18700 $MN_MM_SIZEOF_LINKVAR_DATA) - 1
- Somente podem ser programados índices, de modo que os Bytes endereçados na memória de variáveis de ligação estejam dentro de um limite de formato de dados ⇒
  Índice = n * Bytes, com n = 0, 1, 2, ...
  - `$A_DLB[i]`: i = 0, 1, 2, ...
  - `$A_DLW[i]`: i = 0, 2, 4, ...
  - `$A_DLD[i]`: i = 0, 4, 8, ...
  - `$A_DLR[i]`: i = 0, 8, 16, ...

Exemplo

No sistema de automação estão disponíveis 2 NCUs (NCU1 e NCU2). Na NCU1 está conectado o eixo de máquina AX2, que deve ser deslocado como eixo de lincado da NCU2.

A NCU1 grava ciclicamente o valor real de corrente ($VA_CURR) do eixo AX2 na memória de variáveis de ligação. A NCU2 lê ciclicamente o valor real de corrente transmitido pela comunicação de ligação (Link) e mostra o alarme 61000 no caso de ser ultrapassado o valor limite.

A estrutura de dados na memória de variáveis de ligação está representada na figura a seguir. O valor real de corrente é transferido através do valor REAL.



NCU1
A NCU1 em uma ação síncrona estática, durante o ciclo IPO, grava ciclicamente o valor real de corrente do eixo AX2 na memória de variáveis de ligação através da variável de ligação $A_DLR[16].

Código de programa

```
N111 IDS=1 WHENEVER TRUE DO $A_DLR[16]=$VA_CURR[AX2]
```

NCU2
A NCU2 em uma ação sincronizada estática, durante o ciclo IPO, lê ciclicamente o valor real de corrente do eixo AX2 a partir da memória de variáveis de ligação através da variável de ligação $A_DLR[16]. Se o valor real de corrente for maior que 23.0 A, será exibido o alarme 61000.

Código de programa

```
N222 IDS=1 WHEN $A_DLR[16] > 23.0 DO SETAL(61000)
```

1.1.5 Definição de variáveis de usuário (DEF)

Função

Com o comando `DEF` podemos definir nossas próprias variáveis e preencher as mesmas com valores. Para diferenciar das variáveis de sistema estas são denominadas de variáveis definidas pelo usuário ou de variáveis de usuário (User Data).

De acordo com a área de validade, ou seja, a área em que a variável é visível, existem as seguintes categorias de variáveis de usuário:

- Variáveis de usuário locais (LUD)

  As variáveis de usuário locais (LUD) são variáveis que estão definidas em um programa de peça, que não é o programa principal no momento da execução. Elas são criadas com a chamada do programa de peça e canceladas com o fim do programa de peça ou com NC-Reset. A LUD somente pode ser acessada no contexto do programa de peça em que foi definida.

- Variáveis de usuário globais de programa (PUD)

  As variáveis de usuário globais de programa (PUD) são variáveis que são definidas em um programa de peça utilizado como programa principal. Elas são criadas com o início do programa de peça e canceladas no fim do programa ou com NC-Reset. Os dados PUD podem ser acessados no programa principal e em todas subrotinas.

- Variáveis de usuário globais (GUD)

  As variáveis de usuário globais (GUD) são variáveis globais de NC ou de canal, que estão definidas em um módulo de dados (SGUD, MGUD, UGUD, GUD4 ... GUD9) e que também são preservadas além do Power On. A GUD pode ser acessada em todos os programas de peça.

As variáveis de usuário devem ser definidas antes de serem utilizadas (leitura/gravação). Aqui devem ser observadas as seguintes regras:

- As GUD devem ser definidas em um arquivo de definições, p. ex. _N_DEF_DIR/_M_SGUD_DEF.
- A PUD e a LUD devem ser definidas na parte de definições de um programa de peça.
- A definição de dados deve ser realizada em um bloco próprio.
- Por definição de dados somente pode ser utilizado um tipo de dado.
- Por definição de dados podem ser definidas várias variáveis do mesmo tipo de dado.

Sintaxe

```
DEF <área> <tipo> <parada_pré_processamento> <momento_inicialização>
<unidade_física> <valores_limite> <direitos_acesso>
<nome>[<valor_1>,<valor_2>,<valor_3>]=<valor_inicialização>
```

Significado

`DEF`:	Comando para definição de variáveis de usuário GUD, PUD, LUD	
`<área>`:	Área de validade, apenas relevante para GUD:	
	`NCK`:	Variáveis de usuário globais de NC
	`CHAN`:	Variáveis de usuário globais de canal
`<tipo>`:	Tipo de dado:	
	`INT`:	Valor inteiro com sinal
	`REAL`:	Número real (LONG REAL conforme IEEE)
	`BOOL`:	Valor lógico TRUE (1) / FALSE (0)
	`CHAR`:	Caracteres ASCII
	`STRING[<comprimento_máx>]`:	Sequência de caracteres de tamanho definido
	`AXIS`:	Identificador de eixo/fuso
	`FRAME`:	Especificações geométricas para uma transformação de coordenadas estática
	veja ""	
`<parada_pré_processamento>`:	Parada de pré-processamento, apenas relevante para GUD (opcional)	
	`SYNR`:	Parada de pré-processamento na leitura
	`SYNW`:	Parada de pré-processamento na gravação
	`SYNRW`:	Parada de pré-processamento na leitura e gravação

`<momento_inicialização>`: Momento em que a variável deve ser reinicializada (opcional)

`INIPO`: Power On

`INIRE`: Fim do programa principal, NC-Reset ou Power On

`INICF`: NewConfig ou fim do programa principal, NC-Reset ou Power On

`PRLOC`: Fim do programa principal, NC-Reset após alteração local ou Power On

veja ""

`<unidade_física>`: Unidade física (opcional)

`PHU` <unidade>:

veja ""

`<valores_limite>`: Valor limite inferior e superior (opcional)

`LLI` <valor_limite>: Valor limite inferior (lower limit)

`ULI` <valor_limite>: Valor limite superior (upper limit)

veja ""

`<direitos_acesso>`: Direitos de acesso para leitura / gravação da GUD através de programa de peça ou BTSS (opcional)

`APRP` <nível_proteção>: Leitura: Programa de peça

`APWP` <nível_proteção>: Gravação: Programa de peça

`APRB` <nível_proteção>: Leitura: BTSS

`APWB` <nível_proteção>: Gravação: BTSS

Nível de proteção — Faixa de valores: 0 ... 7

veja ""

`<nome>`: Nome das variáveis

Nota

- Máximo 31 caracteres
- Os primeiros dois caracteres devem ser uma letra e/ou um sublinhado.
- O caractere "$" está reservado para variáveis de sistema e não pode ser utilizado.

`[<valor_1>, <valor_2>, <valor_3>]`: Indicação dos tamanhos de campo para variáveis de campo de 1 até 3 dimensões (opcional)

`<valor_inicialização>`: Valor de inicialização (opcional)

veja ""

Para inicialização de variáveis de campo:

veja ""

Exemplos

Exemplo 1: Definições de variáveis de usuário no módulo de dados para fabricante da máquina

Código de programa

```
%_N_MGUD_DEF                                          ; Módulo GUD: Fabricante da máquina
$PATH=/_N_DEF_DIR
DEF CHAN REAL PHU 24 LLI 0 ULI 10 CORRENTE_1, CORRENTE_2
;  Descrição
;  Definição de duas GUDs: CORRENTE_1, CORRENTE_2
;  Área de validade: Amplo canal
;  Tipo de dado: REAL
;  Parada de pré-processamento: não programada => valor Default = sem parada de pré-processamento
;  Unidade física: 24 = [A]
;  Valores limite: Low = 0.0, High = 10.0
;  Direitos de acesso: não programado => valor Default = 7 = posição do seletor com chave 0
;  Valor de inicialização: não programado => valor Default = 0.0

DEF NCK REAL PHU 13 LLI 10 APWP 3 APRP 3 APWB 0 APRB 2 TEMPO_1=12, TEMPO_2=45
;  Descrição
;  Definição de duas GUDs: TEMPO_1, TEMPO_2
;  Área de validade: Amplo NCK
;  Tipo de dado: REAL
;  Parada de pré-processamento: não programada => valor Default = sem parada de pré-processamento
;  Unidade física: 13 = [s]
;  Valores limite: Low = 10.0, High = não programado => limite superior da faixa de definição
;  Direitos de acesso:
;     Programa de peça: Gravação/leitura = 3 = usuário final
;     BTSS: Gravação = 0 = Siemens, leitura = 3 = usuário final
;  Valor de inicialização: TEMPO_1 = 12.0, TEMPO_2 = 45.0

DEF NCK APWP 3 APRP 3 APWB 0 APRB 3 STRING[5] GUD5_NOME = "COUNTER"
;  Descrição
;  Definição de uma GUD: GUD5_NOME
;  Área de validade: Amplo NCK
;  Tipo de dado: STRING, máx. 5 caracteres
;  Parada de pré-processamento: não programada => valor Default = sem parada de pré-processamento
;  Unidade física: não programada => valor Default = 0 = nenhuma unidade física
;  Valores limite: não programado => limites de faixa de definição: Low = 0, High = 255
;  Direitos de acesso:
;     Programa de peça: Gravação/leitura = 3 = usuário final
;     BTSS: Gravação = 0 = Siemens, leitura = 3 = usuário final
;  Valor de inicialização: "COUNTER"
M30
```

Exemplo 2: Variáveis de usuário globais de programa e locais (PUD / LUD)

Código de programa	**Comentário**
PROC MAIN	; Programa principal
DEF INT VAR1	; Definição PUD
...	
SUB2	; Chamada de subrotina
...	
M30	

Código de programa	**Comentário**
PROC SUB2	; Subrotina SUB2
DEF INT VAR2	; LUD-DEFINITION
...	
IF (VAR1==1)	; Leitura da PUD
VAR1=VAR1+1	; Leitura e gravação da PUD
VAR2=1	; Gravação da LUD
ENDIF	
SUB3	; Chamada de subrotina
...	
M17	

Código de programa	**Comentário**
PROC SUB3	; Subrotina SUB3
...	
IF (VAR1==1)	; Leitura da PUD
VAR1=VAR1+1	; Leitura e gravação da PUD
VAR2=1	; Erro: LUD não reconhecida pela SUB2
ENDIF	
...	
M17	

Exemplo 3: Definição e uso de variáveis de usuário do tipo de dado AXIS

```
Código de programa                          Comentário
DEF AXIS ABSCISSA                           ; 1. Eixo geométrico
DEF AXIS SPINDLE                            ; Fuso
...
IF ISAXIS(1)==FALSE GOTOF CONTINUA
 ABSCISSA = $P_AXN1
CONTINUA:
...
SPINDLE=(S1)                                  1. Fuso
OVRA[SPINDLE]=80                            ; Override de fuso = 80%
SPINDLE=(S3)                                  3. Fuso
```

Condições gerais

Variáveis de usuário globais (GUD)

Durante a definição de variáveis de usuário globais (GUD) devem ser observados os seguintes dados de máquina:

Nº	Identificador: $MN_	Significado
11140	GUD_AREA_ SAVE_TAB	Segurança adicional para módulos GUD
18118 [1)]	MM_NUM_GUD_MODULES	Número de arquivos GUD no sistema de arquivos ativo
18120 [1)]	MM_NUM_GUD_NAMES_NCK	Número de nomes de GUDs globais
18130 [1)]	MM_NUM_GUD_NAMES_CHAN	Número de nomes de GUDs específicas de canal
18140 [1)]	MM_NUM_GUD_NAMES_AXIS	Número de nomes de GUDs específicas de eixo
18150 [1)]	MM_GUD_VALUES_MEM	Capacidade de memória para valores GUD globais
18660 [1)]	MM_NUM_SYNACT_GUD_REAL	Número de GUDs configuráveis de tipo de dado REAL
18661 [1)]	MM_NUM_SYNACT_GUD_INT	Número de GUDs configuráveis de tipo de dado INT
18662 [1)]	MM_NUM_SYNACT_GUD_BOOL	Número de GUDs configuráveis de tipo de dado BOOL
18663 [1)]	MM_NUM_SYNACT_GUD_AXIS	Número de GUDs configuráveis de tipo de dado AXIS
18664 [1)]	MM_NUM_SYNACT_GUD_CHAR	Número de GUDs configuráveis de tipo de dado CHAR
18665 [1)]	MM_NUM_SYNACT_GUD_STRING	Número de GUDs configuráveis de tipo de dado STRING

[1)] Não disponível para o SINUMERIK 828D.

Variáveis de usuário globais de programa (PUD)

ATENÇÃO
Visibilidade de variáveis de usuário locais de programa (PUD) As variáveis de usuário locais de programa (PUD) definidas no programa principal somente estarão visíveis nas subrotinas se o seguinte dado de máquina estiver definido: MD11120 $MN_LUD_EXTENDED_SCOPE = 1 Com MD11120 = 0 as variáveis de usuário locais de programa definidas no programa principal estarão visíveis somente no programa principal.

Utilização por outros canais de uma variável de usuário global de NCK do tipo de dado AXIS

Uma variável de usuário global de NCK do tipo de dado `AXIS`, que foi inicializada com um identificador de eixo pela definição no módulo de dados, somente pode ser utilizada em diferentes canais do NC se o eixo tiver o mesmo número de eixo de canal nestes canais.

Se este não for o caso, a variável deve ser carregada no início do programa de peça ou, como mostra o próximo exemplo, deve ser utilizada a função AXNAME(...) (veja "").

Código de programa	Comentário
`DEF NCK STRING[5] EIXO="X"`	`; Definição em módulo de dados`
`N100 AX[AXNAME(EIXO)]=111 G00`	`; Utilização em programa de peça`

1.1.6 Redefinição de variáveis de sistema, variáveis de usuário e comandos de linguagem NC (REDEF)

Função

Com o comando `REDEF` podem ser alterados os atributos das variáveis de sistema, variáveis de usuário e comandos de linguagem NC. A condição básica para uma redefinição é que ela seja executada a tempo após a definição correspondente.

Em uma redefinição não podem ser alterados vários atributos simultaneamente. Para cada atributo que deve ser alterado deve ser programada uma instrução `REDEF` própria.

Se forem programadas várias alterações concorrentes de atributos, então a última é que sempre estará ativa.

Atributos redefiníveis

Veja "Visão geral dos atributos definíveis e redefiníveis (Página 46)"

Variáveis de usuário locais (PUD / LUD)

Para variáveis de usuário locais (PUD / LUD) não pode ser realizada nenhuma redefinição.

Sintaxe

```
REDEF <nome> <parada_pré_processamento>
REDEF <nome> <unidade_física>
REDEF <nome> <valores_limite>
REDEF <nome> <direitos_acesso>
REDEF <nome> <momento_inicialização>
REDEF <nome> <momento_inicialização> <valor_inicialização>
```

Significado

`REDEF`: Comando para redefinição de um determinado atributo de variáveis de sistema, variáveis de usuário e comandos de linguagem NC

`<nome>`: Nome de uma variável ou de um comando de linguagem NC que já foi definido

`<parada_pré_processamento>`: Parada de pré-processamento

- `SYNR`: Parada de pré-processamento na leitura
- `SYNW`: Parada de pré-processamento na gravação
- `SYNRW`: Parada de pré-processamento na leitura e gravação

`<unidade_física>`: Unidade física

`PHU` **<unidade>**:

veja "Atributo: Unidade física (PHU) (Página 38)"

Nota
Não redefinível para:

- Variáveis de sistema
- Dados de usuário globais (GUD)
- Tipos de dados: `BOOL`, `AXIS`, `STRING`, `FRAME`

`<valores_limite>`: Valor limite inferior e/ou superior

`LLI` **<valor_limite>**: Valor limite inferior (lower limit)

`ULI` **<valor_limite>**: Valor limite superior (upper limit)

veja "Atributo: Valores de limite (LLI, ULI) (Página 37)"

Nota
Não redefinível para:

- Variáveis de sistema
- Dados de usuário globais (GUD)
- Tipos de dados: `BOOL`, `AXIS`, `STRING`, `FRAME`

`<direitos_acesso>`: Direitos de acesso para leitura / gravação através de programa de peça ou BTSS

- `APX` **<nível_proteção>**: Execução: Elemento de linguagem NC
- `APRP` **<nível_proteção>**: Leitura: Programa de peça
- `APWP` **<nível_proteção>**: Gravação: Programa de peça

APRB <nível_proteção>:	Leitura: BTSS
APWB <nível_proteção>:	Gravação: BTSS
Nível de proteção	Faixa de valores: 0 ... 7

veja "Atributo: Direitos de acesso (APR, APW, APRP, APWP, APRB, APWB) (Página 41)"

<momento_inicialização>: Momento em que a variável deve ser reinicializada

INIPO:	PowerOn
INIRE:	Fim do programa principal, NC-Reset ou PowerOn
INICF:	NewConfig ou fim do programa principal, NC-Reset ou PowerOn
PRLOC:	Fim do programa principal, NC-Reset após alteração local ou PowerOn

veja "Atributo: Valor de inicialização (Página 33)"

<valor_inicialização>: Valor de inicialização

Com a redefinição do valor de inicialização sempre deve ser especificado junto um momento de inicialização (veja <momento_inicialização>).

veja "Atributo: Valor de inicialização (Página 33)"

Para inicialização de variáveis de campo:

veja ""

Nota

Não redefinível para:

- Variáveis de sistema, exceto dados de ajuste

Exemplo

Redefinições da variável de sistema $TC_DPC1 no módulo de dados para fabricante da máquina

Código de programa

```
%_N_MGUD_DEF                                          ; Módulo GUD: Fabricante da máquina
$PATH=/_N_DEF_DIR
REDEF $TC_DPC1 APWB 2 APWP 3
REDEF $TC_DPC1 PHU 21
REDEF $TC_DPC1 LLI 0 ULI 200
REDEF $TC_DPC1 INIPO (100, 101, 102, 103)
;  Descrição
;  Direito de acesso para gravação: BTSS = nível de proteção 2, programa de peça = nível de proteção 3
;  Nota
;  Com a utilização de arquivos ACCESS a redefinição dos direitos de acesso do
;  _N_MGUD_DEF deve ser passada para o _N_MACCESS_DEF
;  Unidade física = [ % ]
;  Valores limite: inferior = 0, superior = 200
;  A variável de campo é inicializada com o PowerOn com os quatro valores
M30
```

Condições gerais

Granularidade

Uma redefinição sempre estará relacionada à variável inteira, identificada de forma única através de seu nome. Não é possível, por exemplo, em variáveis de campo atribuir diferentes valores de atributo para elementos de campo individuais.

1.1.7 Atributo: Valor de inicialização

Definição (DEF) de variáveis de usuário

Com a definição pode ser definido um valor de inicialização para as seguintes variáveis de usuário:

- Variáveis de usuário globais (GUD)
- Variáveis de usuário globais de programa (PUD)
- Variáveis de usuário locais (LUD)

Redefinição (REDEF) de variáveis de sistema e de usuário

Com a redefinição pode ser definido um valor de inicialização para as seguintes variáveis:

- Dados de sistema
  - Dados de ajuste
- Dados de usuário
  - Parâmetros R
  - Variável de ação síncrona ($AC_MARKER, $AC_PARAM, $AC_TIMER)
  - GUD de ação síncrona (SYG_xy[ ], com x=R, I, B, A, C, S e y=S, M, U, 4, ..., 9)
  - Parâmetros EPS
  - Dados de ferramenta – OEM
  - Dados de magazine – OEM
  - Variáveis de usuário globais (GUD)

Momento de reinicialização
Com a redefinição pode ser especificado o momento em que a variável é reinicializada, isto é, passa-se novamente para o valor de inicialização:

- `INIPO` (Power On)

  A variável é reinicializada com PowerOn.

- `INIRE` (Reset)

  A variável é reinicializada com NC-Reset, BAG-Reset, fim de programa de peça (`M02` / `M30`) ou PowerOn.

- `INICF` (NewConfig)

  A variável é reinicializada com a solicitação de NewConf através da HMI, comando de programa de peça `NEWCONFIG` ou NC-Reset, BAG-Reset, fim de programa de peça (M02 / M30) ou PowerOn.

- `PRLOC` (alteração local de programa)

  A variável somente é reinicializada com NC-Reset, BAG-Reset ou fim de programa (M02 / M30), se ela for alterada no contexto do atual programa de peça.

  O atributo `PRLOC` somente pode ser utilizado no contexto com dados de ajuste programáveis (veja a tabela a seguir).

Tabelas 1- 1 Dados de ajuste programáveis

Número	Identificador	Comando G [1)]
42000	$SC_THREAD_START_ANGLE	SF
42010	$SC_THREAD_RAMP_DISP	DITS / DITE
42400	$SA_PUNCH_DWELLTIME	PDELAYON
42800	$SA_SPIND_ASSIGN_TAB	SETMS
43210	$SA_SPIND_MIN_VELO_G25	G25
43220	$SA_SPIND_MAX_VELO_G26	G26
43230	$SA_SPIND_MAX_VELO_LIMS	LIMS
43300	$SA_ASSIGN_FEED_PER_REV_SOURCE	FPRAON
43420	$SA_WORKAREA_LIMIT_PLUS	G26
43430	$SA_WORKAREA_LIMIT_MINUS	G25
43510	$SA_FIXED_STOP_TORQUE	FXST
43520	$SA_FIXED_STOP_WINDOW	FXSW
43700	$SA_OSCILL_REVERSE_POS1	OSP1
43710	$SA_OSCILL_REVERSE_POS2	OSP2
43720	$SA_OSCILL_DWELL_TIME1	OST1
43730	$SA_OSCILL_DWELL_TIME2	OST2
43740	$SA_OSCILL_VELO	FA
43750	$SA_OSCILL_NUM_SPARK_CYCLES	OSNSC
43760	$SA_OSCILL_END_POS	OSE
43770	$SA_OSCILL_CTRL_MASK	OSCTRL
43780	$SA_OSCILL_IS_ACTIVE	OS
43790	$SA_OSCILL_START_POS	OSB
1) com este comando G é acionado o dado de ajuste		

Condições gerais

Valor de inicialização: Variáveis de usuário globais (GUD)

- Para variáveis de usuário globais (GUD) com a área de validade NCK somente pode ser definido o INIPO (Power On) como momento de inicialização.
- Para variáveis de usuário globais (GUD) com a área de validade CHAN, além do INIPO (Power On) também o INIRE (Reset) ou o INICF (NewConfig) pode ser definido como momento de inicialização.
- Para variáveis de usuário globais (GUD) com a área de validade CHAN e momento de inicialização INIRE (Reset) ou INICF (NewConfig), com NC-Reset, BAG-Reset e NewConfig somente serão reinicializadas nos canais as variáveis, nas quais foram disparados os eventos mencionados.

Valor de inicialização: Tipo de dado FRAME

Para variáveis do tipo de dado `FRAME` não pode ser especificado nenhum valor de inicialização. As variáveis do tipo de dado `FRAME` sempre são inicializadas de modo implícito com o Frame Default.

Valor de inicialização: Tipo de dado CHAR

Para variáveis do tipo de dado `CHAR` ao invés do código ASCII (0...255), também pode ser programado o respectivo caractere ASCII entre aspas, p. ex. "A"

Valor de inicialização: Tipo de dado STRING

Para variáveis do tipo de dado `STRING` devem ser colocadas aspas na sequência de caracteres, p. ex.: ...= "MAQUINA_1"

Valor de inicialização: Tipo de dado AXIS

Para variáveis do tipo de dado `AXIS` com a forma escrita de endereços ampliada o identificador de eixo deve ser colocado entre parênteses, p. ex.: ...=(X3)

Valor de inicialização: Variável de sistema

Para variáveis de sistema não pode ser definido nenhum valor de inicialização específico de usuário através de redefinição. Os valores de inicialização das variáveis de sistema são estabelecidas pelo sistema. Entretanto, através da redefinição pode ser alterado o momento (`INIRE`, `INICF`) em que a variável é reinicializada.

Valor de inicialização implícito: Tipo de dado AXIS

Para variáveis do tipo de dado `AXIS` são utilizados os seguintes valores de inicialização implícitos:

- Dados de sistema: "primeiro eixo geométrico"
- GUD de ação síncrona (Denominação: SYG_A*), PUD, LUD:
  Identificador de eixo do dado de máquina: MD20082
  $MC_AXCONF_CHANAX_DEFAULT_NAME

Valor de inicialização implícito: Dados de ferramenta e de magazine

Para dados de ferramenta e de magazine podem ser definidos valores de inicialização através do seguinte dado de máquina: MD17520 $MN_TOOL_DEFAULT_DATA_MASK

ATENÇÃO
Sincronização A sincronização de eventos que disparam uma reinicialização de uma variável global com a leitura desta variável em outro ponto, é de responsabilidade exclusiva do usuário / fabricante da máquina.

1.1.8 Atributo: Valores de limite (LLI, ULI)

Um valor limite superior e inferior da faixa de definição somente pode ser especificado para os seguintes tipos de dados:

- INT
- REAL
- CHAR

Definição (DEF) de variáveis de usuário: Valores limite e valores de inicialização implícitos

Se na definição de uma variável de usuário de um dos tipos de dados mencionados acima não for definido nenhum valor de inicialização explícito, a variável é passada para o valor de inicialização implícito do tipo de dado:

- INT: 0
- REAL: 0.0
- CHAR: 0

Se o valor de inicialização implícito estiver fora da faixa de definição definida pelos valores limite programados, a variável é inicializada com o valor de limite que estiver mais próximo do valor de inicialização implícito:

- Valor de inicialização implícito < valor limite inferior (LLI) ⇒
  Valor de inicialização = valor limite inferior
- Valor de inicialização implícito < valor limite superior (ULI) ⇒
  Valor de inicialização = valor limite superior

Exemplos:

Código de programa	**Comentário**
`DEF REAL GUD1`	`; Valor limite inferior = limite da faixa de definição`
	`; Valor limite superior = limite da faixa de definição`
	`; nenhum valor de inicialização programado`
	`; => valor de inicialização implícito = 0.0`
`DEF REAL LLI 5.0 GUD2`	`; Valor limite inferior = 5.0`
	`; Valor limite superior = limite da faixa de definição`
	`; => valor de inicialização = 5.0`
`DEF REAL ULI -5 GUD3`	`; Valor limite inferior = limite da faixa de definição`
	`; Valor limite superior = -5.0`
	`; => valor de inicialização = -5.0`

Redefinição (REDEF) de variáveis de usuário: Valores limite e atuais valores reais

Se com a redefinição dos valores limite uma das variáveis de usuário for alterada de modo que o atual valor real esteja fora da nova faixa de definição, então será emitido um alarme e os valores limite não serão assumidos.

Indicação

Redefinição (REDEF) de variáveis de usuário

Na redefinição dos valores limite de uma variável de usuário deve-se observar para uma modificação consistente dos seguintes valores:

- Valores limite
- Valor real
- Valor de inicialização na redefinição e na reinicialização automática em função do INIPO, INIRE ou INICF

1.1.9 Atributo: Unidade física (PHU)

Uma unidade física somente pode ser especificada para variáveis dos seguintes tipos de dados:

- INT
- REAL

Unidades físicas programáveis (PHU)

A especificação da unidade física é dada como número vírgula fixa: `PHU <unidade>`

Podem ser programadas as seguintes unidades físicas:

<unidade>	Significado	Unidade física
0	Nenhuma unidade física	-
1	Posição linear ou angular 1)2)	[mm], [pol.], [graus]
2	Posição linear 2)	[mm], [pol.]
3	Posição angular	[graus]
4	Velocidade linear ou angular 1)2)	[mm/min], [pol./min], [rpm]
5	Velocidade linear 2)	[mm/min]
6	Velocidade angular	[rpm]
7	Aceleração linear ou angular 1)2)	[m/s^2], [$pol./s^2$], [$rot./s^2$]
8	Aceleração linear 2)	[m/s^2], [$pol./s^2$]
9	Aceleração angular	[$rot./s^2$]
10	Solavanco linear ou angular 1)2)	[m/s^3], [$pol./s^3$], [$rot./s^3$]
11	Solavanco linear 2)	[m/s^3], [$pol./s^3$]
12	Solavanco angular	[$rot./s^3$]
13	Tempo	[s]
14	Amplificação do circuito do controlador de posição	[16.667/s]

<unidade>	Significado	Unidade física
15	Avanço por rotação 2)	[mm/rot.], [pol./rot.]
16	Compensação de temperatura 1)2)	[mm], [pol.]
18	Força	[N]
19	Massa	[kg]
20	Momento de inércia 3)	[kgm^2]
21	Por cento	[%]
22	Frequência	[Hz]
23	Tensão	[V]
24	Corrente	[A]
25	Temperatura	[°C]
26	Ângulo	[graus]
27	KV	[1000/min]
28	Posição linear ou angular 3)	[mm], [pol.], [graus]
29	Velocidade de corte 2)	[m/min], [feet/min]
30	Velocidade periférica 2)	[m/s], [feet/s]
31	Resistência	[Ohm]
32	Indutância	[mH]
33	Torque 3)	[Nm]
34	Constante de torque 3)	[Nm/A]
35	Amplificação do controlador de corrente	[V/A]
36	Amplificação do controlador de rotação 3)	[Nm/(rad*s)]
37	Número de rotações	[rpm]
42	Potência	[kW]
43	Corrente, pequena	[µA]
46	Torque, pequeno 3)	[µNm]
48	Por milha	-
49	-	[Hz/s]
65	Fluxo (vazão)	[l/min]
66	Pressão	[bar]
67	Volume 3)	[cm^3]
68	Amplificação de percurso 3)	[mm/(V*min)]
69	Amplificação de percurso do controlador de força	[N/V]
155	Passo de rosca 3)	[mm/rot.], [pol./rot.]
156	Variação de passo de rosca 3)	[mm/rot. / rot.], [pol./rot. / rot.]
1) A unidade física depende do tipo de eixo: Eixo linear ou eixo rotativo		

<unidade>	Significado	Unidade física
2) Comutação do sistema de medidas G70/G71(polegadas/métrico) Após uma comutação do sistema básico $MN_SCALING_SYSTEM_IS_METRIC) com G70/G71, nos acessos de gravação e leitura das variáveis de sistema e de usuário referentes aos comprimentos (distâncias) não ocorre **nenhuma** conversão dos valores (valor real, valor nominal e valores de limite) G700/G710(polegadas/métrico) Após uma comutação do sistema básico (MD10240 $MN_SCALING_SYSTEM_IS_METRIC), nos acessos de gravação e leitura das variáveis de sistema e de usuário referentes aos comprimentos (distâncias) ocorre **uma** conversão dos valores (valor real, valor nominal e valores de limite)		
3) A variável **não** é convertida automaticamente para o atual sistema de medidas do NC (polegadas/métrico). A conversão é de responsabilidade exclusiva do usuário / fabricante da máquina.		

Indicação

Excesso de níveis através da conversão de formato

O formato interno de armazenamento para todas variáveis de usuário (GUD / PUD / LUD) com unidades físicas de distância é métrico. Um uso excessivo deste tipo de variáveis no processamento principal do NCK, por exemplo em ações síncronas, pode, no caso de uma mudança do sistema de medidas, provocar o estouro do tempo (timeout) de processamento do nível do interpolador, alarme 4240.

ATENÇÃO

Compatibilidade de unidades

Ao serem utilizadas variáveis (atribuição, comparação, cálculos, etc.) não é realizado nenhum controle de compatibilidade das unidades envolvidas. Uma eventual necessidade de conversão é de responsabilidade exclusiva do usuário / fabricante da máquina.

1.1.10 Atributo: Direitos de acesso (APR, APW, APRP, APWP, APRB, APWB)

Os direitos de acesso correspondem aos seguintes níveis de proteção que devem ser informados na programação:

Direito de acesso	Nível de proteção
Senha do sistema	0
Senha do fabricante da máquina	1
Senha da Assistência Técnica	2
Senha do usuário final	3
Seletor com chave na posição 3	4
Seletor com chave na posição 2	5
Seletor com chave na posição 1	6
Seletor com chave na posição 0	7

Definição (DEF) de variáveis de usuário

Os direitos de acesso (APR... / APW...) podem ser definidos para as seguintes variáveis:

- Dados de usuário globais (GUD)

Redefinição (REDEF) de variáveis de sistema e de usuário

Os direitos de acesso (APR... / APW...) podem ser redefinidos para as seguintes variáveis:

- Dados de sistema
  - Dados de máquina
  - Dados de ajuste
  - FRAME
  - Dados de processo
  - Compensação de erro de passo de fuso (EEC)
  - Compensação cruzada (CEC)
  - Compensação de erro de quadrante (QEC)
  - Dados de magazine
  - Dados de ferramenta
  - Áreas de proteção
  - Porta-ferramenta orientável
  - Cadeias cinemáticas
  - Áreas de proteção 3D
  - Limite da área de trabalho
  - Dados de ferramenta ISO

- Dados de usuário
  - Parâmetros R
  - Variável de ação síncrona ($AC_MARKER, $AC_PARAM, $AC_TIMER)
  - GUD de ação síncrona (SYG_xy[ ], com x=R, I, B, A, C, S e y=S, M, U, 4, ..., 9)
  - Parâmetros EPS
  - Dados de ferramenta – OEM
  - Dados de magazine – OEM
  - Variáveis de usuário globais (GUD)

Indicação

Na redefinição o direito de acesso pode ser atribuído livremente a uma variável entre o nível de proteção mais baixo 7 e o próprio nível de proteção, por exemplo 1 (fabricante da máquina).

Redefinição (REDEF) de comandos de linguagem NC

O direito de acesso e de execução (APX) pode ser redefinido para os seguintes comandos de linguagem NC:

- Funções G / Condições de curso

  Literatura:

  /PG/ Manual de programação Fundamentos; Capítulo: Funções G / Condições de curso
- Funções pré-definidas

  Literatura:

  /PG/ Manual de programação Fundamentos; Capítulo: Funções pré-definidas
- Chamadas de subrotina pré-definidas

  Literatura:

  /PG/ Manual de programação Fundamentos; Capítulo: Chamadas de subrotina pré-definidas
- Instrução DO para ações síncronas
- Identificador de programa de ciclos

  O ciclo deve ser armazenado em um diretório de ciclos e conter uma instrução PROC.

Direitos de acesso relacionados aos programas de peça e ciclos (APRP, APWP)

Os diferentes direitos de acesso possuem os seguintes efeitos para o acesso em um programa de peça ou ciclo:

- APRP 0 / APWP 0
  - a senha de sistema deve ser informada para execução do programa de peça
  - o ciclo deve estar armazenado no diretório _N_CST_DIR (sistema)
  - para o diretório _N_CST_DIR o direito de execução deve ser ajustado para sistema no MD11160 $MN_ACCESS_EXEC_CST
- APRP 1 / APWP 1 ou APRP 2 / APWP 2
  - a senha de fabricante de máquina ou de assistência técnica deve ser informada para execução do programa de peça
  - o ciclo deve estar armazenado no diretório _N_CMA_DIR (fabricante da máquina) ou _N_CST_DIR
  - para os diretórios _N_CMA_DIR e _N_CST_DIR os direitos de execução devem estar ajustados pelo menos para fabricante de máquina nos dados de máquina MD11161 $MN_ACCESS_EXEC_CMA e MD11160 $MN_ACCESS_EXEC_CST, respectivamente
- APRP 3 / APWP 3
  - a senha de usuário final deve ser informada para execução do programa de peça
  - o ciclo deve estar armazenado no diretório _N_CUS_DIR (usuário), _N_CMA_DIR ou _N_CST_DIR
  - para os diretórios _N_CUS_DIR, _N_CMA_DIR e _N_CST_DIR os direitos de execução devem estar ajustados pelo menos para usuário final nos dados de máquina MD11162 $MN_ACCESS_EXEC_CUS, MD11161 $MN_ACCESS_EXEC_CMA e MD11160 $MN_ACCESS_EXEC_CST, respectivamente
- APRP 4...7 / APWP 4...7
  - para execução do programa de peça o seletor com chave deve estar ajustado na posição 3 ... 0
  - o ciclo deve estar armazenado no diretório _N_CUS_DIR, _N_CMA_DIR ou _N_CST_DIR
  - para os diretórios _N_CUS_DIR, _N_CMA_DIR e _N_CST_DIR os direitos de execução devem estar ajustados pelo menos de acordo com a posição do seletor com chave nos dados de máquina MD11162 $MN_ACCESS_EXEC_CUS, MD11161 $MN_ACCESS_EXEC_CMA e MD11160 $MN_ACCESS_EXEC_CST, respectivamente

Direitos de acesso relacionados ao BTSS (APRB, APWB)

Os direitos de acesso (APRB, APWB) restringem de modo uniforme o acesso às variáveis de sistema e de usuário através do BTSS para todos componentes do sistema (HMI, PLC, processadores externos, serviços EPS, etc.).

Indicação

Direitos de acesso locais da HMI

Para alterações dos direitos de acesso aos dados de sistema deve-se atentar para que estas sejam realizadas de modo consistente em relação aos direitos de acesso definidos através dos mecanismos da HMI.

Atributos de acesso APR / APW

Por motivos de incompatibilidade os atributos APR e APW são refletidos implicitamente nos atributos APRP / APRB e APWP / APWB:

- APR x ⇒ APRP x APRB x
- APW y ⇒ APWP y APWB y

Ajuste dos direitos de acesso através de arquivos ACCESS

No caso de utilização de arquivos ACCESS para atribuição de direitos de acesso, as redefinições de direitos de acesso para dados de sistema, dados de usuário e comandos de linguagem NC somente poderão ser programadas nestes arquivos ACCESS. Uma exceção são os dados de usuário globais (GUD). Para estes, se necessária, a redefinição dos direitos de acesso ainda pode ser programada nos respectivos arquivos de definição.

Para uma plena proteção de acesso os dados de máquina devem ser adaptados de modo consistente nos respectivos diretórios para os direitos de execução e para a proteção de acesso.

Adota-se o seguinte procedimento básico:

- Criação dos arquivos de definição necessários:
  - _N_DEF_DIR/_N_SACCESS_DEF
  - _N_DEF_DIR/_N_MACCESS_DEF
  - _N_DEF_DIR/_N_UACCESS_DEF
- Parametrização do direito de gravação para os arquivos de definição no valor necessário para a redefinição:
  - MD11170 $MN_ACCESS_WRITE_SACCESS
  - MD11171 $MN_ACCESS_WRITE_MACCESS
  - MD11172 $MN_ACCESS_WRITE_UACCESS

- Para acessar os elementos de ciclos protegidos deve ser realizada a adaptação dos direitos de execução e de gravação dos diretórios de ciclos _N_CST_DIR, _N_CMA_DIR e _N_CST_DIR:

  Direitos de execução

  – MD11160 $MN_ACCESS_EXEC_CST
  – MD11161 $MN_ACCESS_EXEC_CMA
  – MD11162 $MN_ACCESS_EXEC_CUS

  Direitos de gravação

  – MD11165 $MN_ACCESS_WRITE_CST
  – MD11166 $MN_ACCESS_WRITE_CMA
  – MD11167 MN_ACCESS_WRITE_CUS

  O direito de execução deve ser passado pelo menos para o mesmo nível de proteção como o nível de proteção mais alto do elemento utilizado.

  O direito de gravação deve ser passado pelo menos para o mesmo nível de proteção como o direito de execução.

- Os direitos de gravação dos diretórios de ciclos locais da HMI devem ser passados para o mesmo nível de proteção como o dos diretórios de ciclos locais do NC.

  Literatura

  /BAD/ Manual de operação HMI-Advanced,

  Capítulo: Área de operação Serviços > Gerenciar dados > Alterar propriedades

Chamadas de subrotinas em arquivos ACCESS

Para uma expansão da estruturação da proteção de acesso nos arquivos ACCESS também podem ser chamadas subrotinas (extensão SPF ou MPF). Neste caso as subrotinas herdam os direitos de execução do arquivo ACCESS chamado.

Indicação

Nos arquivos ACCESS somente podem ser redefinidos os direitos de acesso. Todos demais atributos ainda devem programados e redefinidos nos respectivos arquivos de definição.

1.1.11 Visão geral dos atributos definíveis e redefiníveis

As seguintes tabelas mostram quais atributos podem ser definidos (DEF) e/ou redefinidos (REDEF) em quais tipos de dados.

Dados do sistema

Tipo de dados	Valor inic.	Valores limite	Unidade física	Direitos de acesso
Dados de máquina	---	---	---	REDEF
Dados de ajuste	REDEF	---	---	REDEF
Dados de FRAME	---	---	---	REDEF
Dados de processo	---	---	---	REDEF
Comp. de erro de passo de fuso (EEC)	---	---	---	REDEF
Compensação cruzada (CEC)	---	---	---	REDEF
Compensação de erro de quadrante (QEC)	---	---	---	REDEF
Dados de magazine	---	---	---	REDEF
Dados de ferramenta	---	---	---	REDEF
Áreas de proteção	---	---	---	REDEF
Porta-ferramenta orientável	---	---	---	REDEF
Cadeias cinemáticas	---	---	---	REDEF
Áreas de proteção 3D	---	---	---	REDEF
Limite de área de trabalho	---	---	---	REDEF
Dados de ferramenta ISO	---	---	---	REDEF

Dados de usuário

Tipo de dados	Valor inic.	Valores limite	Unidade física	Direitos de acesso
Parâmetros R	REDEF	REDEF	REDEF	REDEF
Variável de ação síncrona ($AC_...)	REDEF	REDEF	REDEF	REDEF
GUD de ação síncrona (SYG_...)	REDEF	REDEF	REDEF	REDEF
Parâmetros EPS	REDEF	REDEF	REDEF	REDEF
Dados de ferramenta – OEM	REDEF	REDEF	REDEF	REDEF
Dados de magazine – OEM	REDEF	REDEF	REDEF	REDEF
Variáveis de usuário globais (GUD)	DEF / REDEF	DEF	DEF	DEF / REDEF
Variáveis de usuários locais (PUD / LUD)	DEF	DEF	DEF	---

1.1.12 Definição e inicialização de variáveis de campo (DEF, SET, REP)

Função

Uma variável de usuário pode ser definida como campo (Array) de 1 até 3 dimensões:

- 1 dimensão: `DEF <tipo de dado> <nome da variável>[<n>]`
- 2 dimensões: `DEF <tipo de dado> <nome da variável>[<n>,<m>]`
- 3 dimensões: `DEF <tipo de dado> <nome da variável>[<n>,<m>,<o>]`

Indicação

As variáveis de usuário do tipo de dados STRING podem ser definidas no máximo como campo de 2 dimensões.

Tipos de dados

As variáveis de usuário podem ser definidas como campos para os seguintes tipos de dados: BOOL, CHAR, INT, REAL, STRING, AXIS, FRAME

Atribuição de valores nos elementos de campo

As atribuições de valores nos elementos de campo podem ser realizadas nas seguintes ocasiões:

- Para a definição de campo (valores de inicialização)
- Durante a execução do programa

Aqui a atribuição de valores pode ser realizada através de:

- Especificação explícita de um elemento de campo
- Especificação explícita de um elemento de campo como elemento de partida e especificação de uma lista de valores (`SET`)
- Especificação explícita de um elemento de campo como elemento de partida e especificação de um valor e a freqüência de sua repetição (`REP`)

Indicação

As variáveis de usuário do tipo de dados FRAME não podem ser atribuídas com valores de inicialização.

Sintaxe (`DEF`)

```
DEF <tipo de dado> <nome da variável>[<n>,<m>,<o>]
DEF  STRING[<tamanho de String>] <nome de variável>[<n>,<m>]
```

Sintaxe (DEF...=SET...)

Utilização de uma lista de valores:

- Para a definição:

```
DEF <tipo de dado> <nome de variável>[<n>,<m>,<o>] = SET(<valor1>,<valor2>,...)
```

Equivalente com:

```
DEF <tipo de dado> <nome de variável>[<n>,<m>,<o>] = (<valor1>,<valor2>,...)
```

Indicação

A especificação do SET é opcional para a inicialização através de uma lista de valores .

- Para uma atribuição de valores:

```
<nome da variável>[<n>,<m>,<o>]=SET(<VALOR1>,<valor2>,...)
```

Sintaxe (DEF...=REP...)

Utilização de um valor com repetição

- Para a definição:

```
DEF <tipo de dado> <nome da variável>[<n>,<m>,<o>]=REP(<valor>)
```

```
DEF <tipo de dado> <nome de variável>[<n>,<m>,<o>] =
REP(<valor>,<número_elementos_de_campo>)
```

- Para uma atribuição de valores:

```
<nome da variável>[<n>,<m>,<o>]=REP(<valor>)
<nome da variável>[<n>,<m>,<o>]=REP(<valor>,<número_de_elmentos_de_campo>)
```

Significado

DEF:	Comando para definição de variáveis
<tipo de dado>:	Tipo de dados das variáveis Faixa de valores: • Para variáveis de sistema: BOOL, CHAR, INT, REAL, STRING, AXIS • Para variáveis GUD ou LUD: BOOL, CHAR, INT, REAL, STRING, AXIS, FRAME
<tamanho de String>:	Número máximo de caracteres para o tipo de dado STRING
<nome de variável>:	Nome de variável
[<n>,<m>,<o>]:	Tamanhos de campo e índices de campo
<n>:	Tamanho e índice de campo para 1ª dimensão Tipo: INT (também AXIS em variáveis de sistema) Faixa de valores: Tamanho de campo máx.: 65535 Índice de campo: 0 ≤ n ≤ 65534

`<m>`: Tamanho e índice de campo para 2ª dimensão

Tipo: INT (também AXIS em variáveis de sistema)

Faixa de valores: Tamanho de campo máx.: 65535
Índice de campo: 0 ≤ m ≤ 65534

`<o>`: Tamanho e índice de campo para 3ª dimensão

Tipo: INT (também AXIS em variáveis de sistema)

Faixa de valores: Tamanho de campo máx.: 65535
Índice de campo: 0 ≤ o ≤ 65534

`SET`: Atribuição de valores através da lista de valores indicada

`(<valor1>,<valor2>,...)`: Lista de valores

`REP`: Atribuição de valores através do `<valor>` indicado

`<valor>`: Valor com que os elementos de campo devem ser escritos na inicialização com `REP`.

`<número_de_elementos_de_campo>`: Número de elementos de campo que devem ser preenchidos com o `<valor>` indicado. Para os elementos de campo restantes a aplicação depende do momento:

- Inicialização na definição de campo:

  → Os elementos de campo restantes são preenchidos com zero
- Atribuição durante a execução do programa:

  → Os atuais valores dos elementos de campo permanecem inalterados.

Se o parâmetro não estiver programado, todos elementos de campo são preenchidos com o `<valor>`.

Se o parâmetro for igual a zero, a aplicação depende do momento:

- Inicialização na definição de campo:

  → Todos elementos são ocupados previamente com zero
- Atribuição durante a execução do programa:

  → Os atuais valores dos elementos de campo permanecem inalterados.

Índice de campo

A ordem implícita dos elementos de campo, por exemplo no caso de uma atribuição de valores através de SET ou REP, é realizada através da iteração do índice de campo da direita para a esquerda.

Exemplo: Inicialização de um campo de 3 dimensões com 24 elementos de campo:

```
DEF INT CAMPO[2,3,4] = REP(1,24)
  CAMPO[0,0,0] = 1          1º Elemento de campo
  CAMPO[0,0,1] = 1          2º Elemento de campo
  CAMPO[0,0,2] = 1          3º Elemento de campo
  CAMPO[0,0,3] = 1          4º Elemento de campo
  ...
  CAMPO[0,1,0] = 1          5º Elemento de campo
  CAMPO[0,1,1] = 1          6º Elemento de campo
  ...
  CAMPO[0,2,3] = 1          12º Elemento de campo
  CAMPO[1,0,0] = 1          13º Elemento de campo
  CAMPO[1,0,1] = 1          14º Elemento de campo
  ...
  CAMPO[1,2,3] = 1          24º Elemento de campo
```

de acordo com:

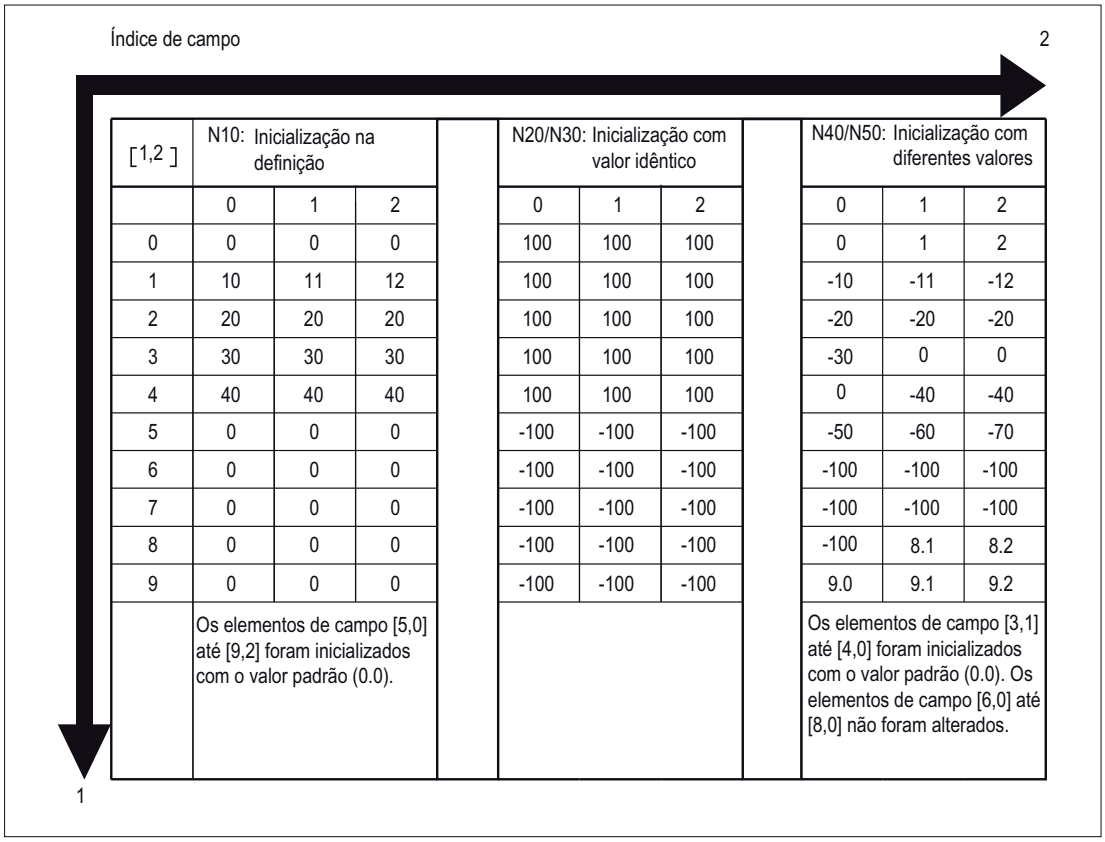
```
FOR n=0 TO 1
  FOR m=0 TO 2
    FOR o=0 TO 3
      CAMPO[n,m,o] = 1
    ENDFOR
  ENDFOR
ENDFOR
```

Exemplo: Inicialização de campos de variáveis completos

Veja a atual ocupação na figura.

Código de programa

```
N10 DEF REAL CAMPO1[10,3]=SET(0,0,0,10,11,12,20,20,20,30,30,30,40,40,40,)
N20 CAMPO1[0,0]=REP(100)
N30 CAMPO1[5,0]=REP(-100)
N40 CAMPO1[0,0]=SET(0,1,2,-10,-11,-12,-20,-20,-20,-30, , , ,-40,-40,-50,-60,-70)
N50 CAMPO1[8,1]=SET(8.1,8.2,9.0,9.1,9.2)
```



[1,2]	N10: Inicialização na definição			N20/N30: Inicialização com valor idêntico			N40/N50: Inicialização com diferentes valores		
	0	1	2	0	1	2	0	1	2
0	0	0	0	100	100	100	0	1	2
1	10	11	12	100	100	100	-10	-11	-12
2	20	20	20	100	100	100	-20	-20	-20
3	30	30	30	100	100	100	-30	0	0
4	40	40	40	100	100	100	0	-40	-40
5	0	0	0	-100	-100	-100	-50	-60	-70
6	0	0	0	-100	-100	-100	-100	-100	-100
7	0	0	0	-100	-100	-100	-100	-100	-100
8	0	0	0	-100	-100	-100	-100	8.1	8.2
9	0	0	0	-100	-100	-100	9.0	9.1	9.2
	Os elementos de campo [5,0] até [9,2] foram inicializados com o valor padrão (0.0).						Os elementos de campo [3,1] até [4,0] foram inicializados com o valor padrão (0.0). Os elementos de campo [6,0] até [8,0] não foram alterados.		

1.1.13 Definição e inicialização de variáveis de campo (DEF, SET, REP): Outras informações

Outras informações (SET)

Inicialização na definição

- Iniciando-se pelo 1º elemento de campo, são inicializados tantos elementos de campo com os valores da lista de valores como foram programados elementos na lista de valores.
- Os elementos de campo sem valores indicados de modo explícito na lista de peças (brechas na lista de valores) são preenchidos com 0.
- Para variáveis do tipo de dado AXIS não são permitidas brechas na lista de valores.
- Se a lista de valores contém mais valores como do que elementos de campo definidos, aparecerá um alarme.

Atribuição de valores na execução do programa

Para a atribuição de valores na execução do programa são aplicadas as regras descritas acima para a definição. Além disso existem as seguintes possibilidades:

- Como elementos na lista de valores também é permitido o uso de expressões.
- A atribuição de valores inicia com o índice de campo programado. Dessa forma os campos parciais são ocupados de modo controlado.

Exemplo:

Código de programa	**Comentário**
`DEF INT CAMPO[5,5]`	`; Definição de campos`
`CAMPO[0,0]=SET(1,2,3,4,5)`	`; Atribuição de valores nos primeiros 5 elementos de campo [0,0] - [0,4]`
`CAMPO[0,0]=SET(1,2, , ,5)`	`; Atribuição de valores com brecha nos primeiros 5 elementos de campo [0,0] - [0,4], elementos de campo [0,2] e [0,3] = 0`
`CAMPO[2,3]=SET(VARIAVEL,4*5.6)`	`; Atribuição de valores com variável e expressão a partir do índice de campo [2,3]: [2,3] = VARIAVEL [2,4] = 4 * 5.6 = 22.4`

Outras informações (REP)

Inicialização na definição

- Todos os elementos ou a quantidade opcional indicada de elementos de campo é iniciada com o valor (constante) indicado.
- As variáveis do tipo de dado FRAME não podem ser inicializadas.

Exemplo:

Código de programa	Comentário
`DEF REAL varName[10]=REP(3.5,4)`	`; Definição de campo e inicialização de elementos de campo [0] até [3] com o valor 3,5`

Atribuição de valores na execução do programa

Para a atribuição de valores na execução do programa são aplicadas as regras descritas acima para a definição. Além disso existem as seguintes possibilidades:

- Como elementos na lista de valores também é permitido o uso de expressões.
- A atribuição de valores inicia com o índice de campo programado. Dessa forma os campos parciais são ocupados de modo controlado.

Exemplos:

Código de programa	Comentário
`DEF REAL varName[10]`	`; Definição de campos`
`varName[5]=REP(4.5,3)`	`; Elementos de campo [5] até [7] = 4,5`
`R10=REP(2.4,3)`	`; Parâmetros R10 até R12 = 2,4`
`DEF FRAME FRM[10]`	`; Definição de campos`
`FRM[5]=REP(CTRANS(X,5))`	`; Elementos de campo [5] até [9] = CTRANS(X,5)`

Outras informações (geral)

Atribuição de valores em dados de máquina axiais

Em princípio os dados de máquina axiais possuem um índice de campo do tipo de dado AXIS. Este índice de campo é ignorado e não processado para atribuição de valores em um dado de máquina axial através do SET ou REP.

Exemplo: Atribuição de valores no dado de máquina MD36200 $MA_AX_VELO_LIMIT

```
$MA_AX_VELO_LIMIT[1,AX1]=SET(1.1, 2.2, 3.3)
```

Corresponde:

```
$MA_AX_VELO_LIMIT[1,AX1]=1.1
$MA_AX_VELO_LIMIT[2,AX1]=2.2
$MA_AX_VELO_LIMIT[3,AX1]=3.3
```

ATENÇÃO

Atribuição de valores em dados de máquina axiais

O índice de campo do tipo de dado AXIS é ignorado e não processado para atribuições de valores em dados de máquina axiais através de SET ou REP.

Necessidade de memória

Tipo de dado	Necessidade de memória por elemento
BOOL	1 Byte
CHAR	1 Byte
INT	4 Bytes
REAL	8 Bytes
STRING	(tamanho de String + 1) Bytes
FRAME	~ 400 Bytes, em função do número de eixos
AXIS	4 Bytes

1.1.14 Tipos de dados

No NC estão disponíveis os seguintes tipos de dados:

Tipo de dado	Significado	Faixa de valores
INT	Valor inteiro com sinal	-2147483648 ... +2147483647
REAL	Número real (LONG REAL conforme IEEE)	±(~2,2*10^{-308} … ~1,8*10^{+308})
BOOL	Valor lógico TRUE (1) e FALSE (0)	1, 0
CHAR	Caracteres ASCII	Código ASCII 0 ... 255
STRING	Sequência de caracteres de tamanho definido	máximo 200 caracteres (sem caractere especial)
AXIS	Identificador de eixo/fuso	Identificador de canal
FRAME	Indicações geométricas para uma transformação estática de coordenadas (deslocamento, rotação, escalonamento, espelhamento)	---

Mudanças de tipo de dado implícitas

As seguintes mudanças de tipo de dado são possíveis e são realizadas de forma implícita nas atribuições e transferências de parâmetros:

de ↓/ para →	REAL	INT	BOOL
REAL	x	o	&
INT	x	x	&
BOOL	x	x	x

x: Possível sem restrições

o: Possível perda de dados ao exceder a faixa de valores ⇒ Alarme;
Arredondamento: Valor após a vírgula ≥ 0,5 ⇒ arredonda para cima, valor após a vírgula < 0,5 ⇒ arredonda para baixo

&: Valor ≠ 0 ⇒ TRUE, valor == 0 ⇒ FALSE

1.2 Programação indireta

1.2.1 Programação indireta de endereços

Função

Na programação indireta de endereços o endereço ampliado (índice) é substituído por uma variável de tipo adequado.

Indicação

A programação indireta de endereços não é possível com:

- N (número de bloco)
- L (subrotina)
- Endereços ajustáveis
  (p. ex. não é permitido X[1] no lugar de X1)

Sintaxe

```
<ENDEREÇO>[<índice>]
```

Significado

`<ENDEREÇO>[...]`:	Endereço fixo com extensão (índice)
`<índice>`:	Variável, p. ex. para número de fuso, eixo, ...

Exemplos

Exemplo 1: Programação indireta de um número de fuso

Programação direta:

Código de programa	Comentário
`S1=300`	`; Rotação de 300 rpm para o fuso de número 1.`

Programação indireta:

Código de programa	Comentário
`DEF INT SPINU=1`	`; Definição das variáveis do tipo INT e atribuição de valores.`
`S[SPINU]=300`	`; Rotação de 300 rpm para o fuso cujo número está armazenado na variável SPINU (no exemplo temos o fuso de número 1).`

Exemplo 2: Programação indireta de um eixo

Programação direta:

Código de programa	Comentário
FA[U]=300	; Avanço 300 para o eixo "U".

Programação indireta:

Código de programa	Comentário
DEF AXIS AXVAR2=U	; Definição de uma variável do tipo AXIS e atribuição de valores.
FA[AXVAR2]=300	; Avanço 300 para o eixo cujo nome de endereço está armazenado na variável de nome AXVAR2.

Exemplo 3: Programação indireta de um eixo

Programação direta:

Programação	Comentário
$AA_MM[X]	; Ler o valor de medição do apalpador de medição (MCS) do eixo "X".

Programação indireta:

Código de programa	Comentário
DEF AXIS AXVAR3=X	; Definição de uma variável do tipo AXIS e atribuição de valores.
$AA_MM[AXVAR3]	; Ler o valor de medição do apalpador de medição (MCS) do eixo cujo nome está armazenado na variável AXVAR3.

Exemplo 4: Programação indireta de um eixo

Programação direta:

Código de programa
X1=100 X2=200

Programação indireta:

Código de programa	Comentário
DEF AXIS AXVAR1 AXVAR2	; Definição de duas variáveis do tipo AXIS.
AXVAR1=(X1) AXVAR2=(X2)	; Atribuição dos nomes de eixo.
AX[AXVAR1]=100 AX[AXVAR2]=200	; Deslocamento dos eixos cujos nomes de endereço estão armazenados nas variáveis de nome AXVAR1 e AXVAR2.

Exemplo 5: Programação indireta de um eixo

Programação direta:

Código de programa
G2 X100 I20

Programação indireta:

Código de programa	Comentário
DEF AXIS AXVAR1=X	; Definição de uma variável do tipo AXIS e atribuição de valores.
G2 X100 IP[AXVAR1]=20	; Programação indireta da indicação do centro para o eixo cujo nome de endereço está armazenado na variável de nome AXVAR1.

Exemplo 6: Programação indireta de elementos de campo

Programação direta:

Código de programa	Comentário
DEF INT CAMPO1[4,5]	; Definição do campo 1.

Programação indireta:

Código de programa	Comentário
DEFINE DIM1 AS 4	; Em dimensões de campo os tamanhos de campo devem ser especificados como valores fixos.
DEFINE DIM2 AS 5	
DEF INT CAMPO[DIM1,DIM2]	
CAMPO[DIM1-1,DIM2-1]=5	

Exemplo 7: Chamada indireta de subrotina

Código de programa	Comentário
CALL "L" << R10	; Chamada do programa cujo número está no R10 (encadeamento de string).

1.2.2 Programação indireta de códigos G

Função

A programação indireta de códigos G possibilita uma programação efetiva de ciclos.

Sintaxe

```
G[<grupo>]=<número>
```

Significado

`G[...]`:	Comando G com extensão (índice)
`<grupo>`:	Parâmetro de índice: Grupo de funções G Tipo: INT
`<número>`:	Variável para o número de código G Tipo: INT ou REAL

Indicação

Geralmente somente podem ser programados de forma indireta os códigos G que não são determinantes na sintaxe.

Dos códigos G que são determinantes na sintaxe, apenas são possíveis aqueles do grupo 1 de funções G.
Os códigos G determinantes de sintaxe dos grupos 2. 3 e 4 de funções G não são aplicáveis.

Indicação

Na programação indireta de códigos G não é permitida nenhuma função aritmética. Um cálculo necessário do número de código G devem ser realizado em uma linha própria do programa de peça, ainda antes da programação indireta do código G.

Exemplos

Exemplo 1: Deslocamento de ponto zero ajustável (grupo de funções G 8)

```
Código de programa                          Comentário
N1010 DEF INT INT_VAR
N1020 INT_VAR=2
...
N1090 G[8]=INT_VAR G1 X0 Y0                 ; G54
N1100 INT_VAR=INT_VAR+1                     ; Cálculo de código G
N1110 G[8]=INT_VAR G1 X0 Y0                 ; G55
```

Exemplo 2: Seleção de plano (grupo de funções G 6)

```
Código de programa                    Comentário
N2010 R10=$P_GG[6]                    ; Ler a função G ativa do grupo de funções
                                        G 6
...
N2090 G[6]=R10
```

Literatura

Para informações sobre os grupos de função G, veja:
Manual de programação Fundamentos; Capítulo "Grupos de funções G"

1.2.3 Programação indireta de atributos de posição (GP)

Função

Os atributos de posição, como a programação incremental ou absoluta da posição do eixo, podem ser programados indiretamente como variáveis em conjunto com a palavra-chave `GP`.

Aplicação

A programação indireta de atributos de posição tem aplicação nos **ciclos de substituição**, tendo a seguinte vantagem sobre a programação de atributos de posição como palavra-chave (p. ex. `IC`, `AC`, ...):

Através da programação indireta como variáveis **não** será necessária nenhuma instrução `CASE` que se bifurca através de todos possíveis atributos de posição.

Sintaxe

```
<COMANDO_POSICIONAMENTO>[<eixo/fuso>]=
GP(<posição>,<atributo de posição)
<eixo/fuso>=GP(<posição>,<atributo de posição>)
```

Significado

<COMANDO_POSICIONAMENTO>[]:	Os seguintes comandos de posicionamento podem ser programados juntos com a palavra-chave GP: POS, POSA, SPOS, SPOSA Além disso também é possível: • todos identificadores de eixo/fuso disponíveis no canal: <eixo/fuso> • identificador variável de eixo/fuso AX
<eixo/fuso>:	Eixo/fuso que deve ser posicionado
GP():	Palavra-chave para posicionamento
<posição>:	Parâmetro 1 Posição de eixo/fuso como constante ou variável
<atributo de posição>:	Parâmetro 2 Atributo de posição (p. ex. modo de aproximação da posição) como variável (p. ex. $P_SUB_SPOSMODE) ou como palavra-chave (IC, AC, ...)

Os valores retornados das variáveis possuem o seguinte significado:

Valor	Significado	Permitido para:
0	Nenhuma alteração do atributo de posição	
1	AC	POS, POSA, SPOS, SPOSA, AX, endereço de eixo
2	IC	POS, POSA, SPOS, SPOSA, AX, endereço de eixo
3	DC	POS, POSA, SPOS, SPOSA, AX, endereço de eixo
4	ACP	POS, POSA, SPOS, SPOSA, AX, endereço de eixo
5	ACN	POS, POSA, SPOS, SPOSA, AX, endereço de eixo
6	OC	-
7	PC	-
8	DAC	POS, POSA, AX, endereço de eixo
9	DIC	POS, POSA, AX, endereço de eixo
10	RAC	POS, POSA, AX, endereço de eixo
11	RIC	POS, POSA, AX, endereço de eixo
12	CAC	POS, POSA
13	CIC	POS, POSA
14	CDC	POS, POSA
15	CACP	POS, POSA
16	CACN	POS, POSA

Exemplo

Em um acoplamento ativo de fusos sincronizados entre o fuso mestre S1 e o fuso escravo S2, através do comando `SPOS` no programa principal é realizada a chamada do seguinte ciclo de substituição para posicionamento dos fusos.

O posicionamento é realizado através da instrução no `N2230`:
`SPOS[1]=GP($P_SUB_SPOSIT,$P_SUB_SPOSMODE)`
`SPOS[2]=GP($P_SUB_SPOSIT,$P_SUB_SPOSMODE)`

A posição a ser aproximada é lida da variável de sistema $P_SUB_SPOSIT, o modo de aproximação da posição da variável de sistema $P_SUB_SPOSMODE.

```
Código de programa                                    Comentário
N1000 PROC LANG_SUB DISPLOF SBLOF
...
N2100 IF($P_SUB_AXFCT==2)
N2110                                                 ;  Substituição do comando SPOS / SPOSA / M19 no
                                                         acoplamento de fusos sincronizados
N2185 DELAYFSTON                                      ;  Início da área stop-delay
N2190 COUPOF(S2,S1)                                   ;  Desativar o acoplamento de fusos
                                                         sincronizados
N2200                                                 ;  Posicionar fuso mestre e fuso escravo
N2210 IF($P_SUB_SPOS==TRUE) OR ($P_SUB_SPOSA==TRUE)
N2220                                                 ;  Posicionar fuso com SPOS:
N2230    SPOS[1]=GP($P_SUB_SPOSIT,$P_SUB_SPOSMODE)
         SPOS[2]=GP($P_SUB_SPOSIT,$P_SUB_SPOSMODE)
N2250 ELSE
N2260                                                 ;  Posicionar fuso com M19:
N2270 M1=19 M2=19                                     ;  Posicionar fuso mestre e fuso escravo
N2280 ENDIF
N2285 DELAYFSTOF                                      ;  Fim da área stop-delay
N2290 COUPON(S2,S1)                                   ;  Ativar acoplamento de fusos sincronizados
N2410 ELSE
N2420                                                 ;  Consulta quanto a outras substituições
...
N3300 ENDIF
...
N9999 RET
```

Condições gerais

- A programação indireta de atributos de posição não é possível em ações sincronizadas.

Literatura

Manual de funções básicas; BAG, Canal, Operação do programa, Comportamento Reset (K1),
Capítulo: Substituição de funções NC através de subrotinas

1.2.4 Programação indireta de linhas de programa de peça (EXECSTRING)

Função

Com o comando de programa de peça EXECSTRING é possível executar uma variável de String, que foi gerada anteriormente, como uma linha do programa de peça.

Sintaxe

EXECSTRING é programado em uma linha de programa de peça própria:
`EXECSTRING(<variável String>)`

Significado

EXECSTRING:	Comando para execução de uma variável de String como linha de programa de peça
<variável String>:	Variável do tipo STRING que contém a própria linha de programa de peça que deve ser executada

Indicação

Com o EXECSTRING podem ser descartadas todas as construções de programa de peça que foram programadas em uma parte de um programa de peça. Excluídas disso estão as instruções PROC e DEF, bem como o uso frequente em arquivos INI e DEF.

Exemplo

```
Código de programa                          Comentário
N100 DEF STRING[100] BLOCO                  ; Definição das variáveis de String para
                                              gravar as linhas de programa de peça
                                              que devem ser executadas.
N110 DEF STRING[10] MFCT1="M7"
...
N200 EXECSTRING(MFCT1 << "M4711")           ; Execução da linha de programa de peça
                                              "M7 M4711".
...
N300 R10=1
N310 BLOCO="M3"
N320 IF(R10)
N330 BLOCO = BLOCO << MFCT1
N340 ENDIF
N350 EXECSTRING(BLOCO)                      ; Execução da linha de programa de peça
                                              "M3 M7"
```

1.3 Funções de cálculo

Função

As funções de cálculo são aplicadas com prioridade sobre os parâmetros R e as variáveis (ou constantes e funções) do tipo REAL. Também são permitidos os tipos INT e CHAR.

Operador / Função de cálculo	Significado
+	Adição
-	Subtração
*	Multiplicação
/	Divisão **Atenção**: (tipo INT)/(tipo INT)=(tipo REAL); exemplo: 3/4 = 0.75
DIV	Divisão, para tipo de variável INT e REAL **Atenção**: (tipo INT)DIV(tipo INT)=(tipo INT); exemplo: 3 DIV 4 = 0
MOD	A divisão Modulo (apenas para tipo INT) retorna o resto de uma divisão INT Exemplo: 3 MOD 4 = 3
:	Operador de concatenação (para variáveis FRAME)
Sin()	Seno
COS()	Coseno
TAN()	Tangente
ASIN()	Arco seno
ACOS()	Arco coseno
ATAN2(,)	Arco tangente2
SQRT()	Raiz quadrada
ABS()	Valor
POT()	2. Potência (quadrada)
TRUNC()	Parte inteira Precisão em comandos de comparação ajustável com TRUNC (veja "Correção da precisão em erros de comparação (TRUNC) (Página 69)")
ROUND()	Arredondamento para um número inteiro
LN()	Logaritmo natural
EXP()	Função exponencial
MINVAL()	menor valor de duas variáveis (veja "Mínimo, máximo e área de variáveis (MINVAL, MAXVAL, BOUND) (Página 71)")

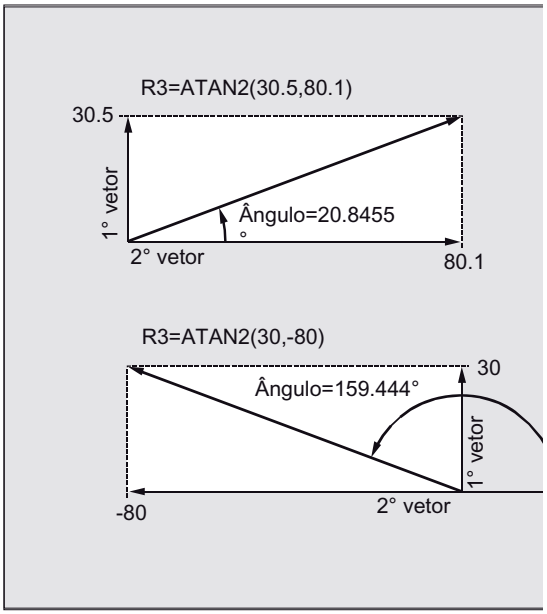
`MAXVAL()`	Maior valor de duas variáveis (veja "Mínimo, máximo e área de variáveis (MINVAL, MAXVAL, BOUND) (Página 71)")
`BOUND()`	Valor de variável que está na faixa de valores definida (veja "Mínimo, máximo e área de variáveis (MINVAL, MAXVAL, BOUND) (Página 71)")
`CTRANS()`	Deslocamento
`CROT()`	Rotação
`CSCALE()`	Alteração de escala
`CMIRROR()`	Espelhamento

Programação

Nas funções de cálculo se aplica a forma escrita usual de matemática. As prioridades na execução são definidas através de parênteses. Para as funções trigonométricas e suas funções inversas se aplica a indicação em graus (ângulo reto = 90°).

Exemplos

Exemplo 1: ATAN2



A função de cálculo ATAN2 calcula o ângulo do vetor formado por dois vetores posicionados perpendicularmente.

O resultado está na área de quatro quadrantes (-180° < 0 < +180°).

A base para a referência angular sempre é o 2º valor no sentido positivo.

Exemplo 2: Inicialização de campos de variáveis completos

Código de programa	**Comentário**
`R1=R1+1`	`; Novo R1 = antigo R1 +1`
`R1=R2+R3 R4=R5-R6 R7=R8*R9`	
`R10=R11/R12 R13=SIN(25.3)`	
`R14=R1*R2+R3`	`; O cálculo com pontos tem prioridade sobre o cálculo com traços.`
`R14=(R1+R2)*R3`	`; Parênteses/colchetes são calculados primeiro.`
`R15=SQRT(POT(R1)+POT(R2))`	`; Parênteses internos são ativados primeiro: R15 = raiz quadrada de (R1+R2)`
`RESFRAME=FRAME1:FRAME2` `FRAME3=CTRANS(…):CROT(…)`	`; Com o operador de concatenação os Frames são ligados a um Frame resultante ou atribuídos aos valores de componente de Frame.`

1.4 Operadores de comparação e operadores lógicos

Função

Operações de comparação podem ser usadas, por exemplo, para formulação de uma condição de salto. Também podem ser comparadas expressões complexas.

As operações de comparação são aplicáveis para variáveis do tipo CHAR, INT, REAL e BOOL. No tipo CHAR é comparado o valor do código.
Nos tipos STRING, AXIS e FRAME são possíveis: == e <>, que também podem ser aplicados em operações do tipo STRING em ações sincronizadas.

O resultado das operações comparadas sempre será do tipo BOOL.

Operadores lógicos servem para ligação de valores lógicos.

As operações lógicas somente podem ser aplicadas em variáveis do tipo BOOL. Através da conversão interna de tipos elas também podem ser aplicadas nos tipos de dados CHAR, INT e REAL.

Nas operações lógicas (booleanas) aplica-se os tipos de dados BOOL, CHAR, INT e REAL:

- 0 corresponde a: FALSE
- diferente de 0 corresponde a: TRUE

Operadores lógicos por Bits

Com as variáveis do tipo CHAR e INT também podem ser empregados operadores lógicos por Bits. Eventualmente é realizada uma conversão automática de tipos.

Programação

Operador de comparação	Significado
==	igual
<>	diferente
>	maior
<	menor
>=	maior ou igual
<=	menor ou igual

Operador lógico	Significado
AND	E
OR	OU
NOT	Negação
XOR	OU exclusivo

Operador lógico por Bits	Significado
B_AND	E bit a bit
B_OR	OU bit a bit
B_NOT	Negação bit a bit
B_XOR	OU exclusivo bit a bit

Indicação

Em expressões aritméticas se pode definir a ordem de execução através de parênteses, e com isso, fugir das regras convencionais de prioridade.

Indicação

Entre operandos e operadores BOOLEANOS devem ser escritos espaços.

Indicação

O operador B_NOT refere-se apenas a um operando. Este se encontra após o operador.

Exemplos

Exemplo 1: Operadores de comparação

```
IF R10>=100 GOTOF DESTINO
```

ou

```
R11=R10>=100
IF R11 GOTOF DESTINO
```

O resultado da comparação R10>=100 primeiro é armazenado temporariamente no R11.

Exemplo 2: Operadores lógicos

```
IF (R10<50) AND ($AA_IM[X]>=17.5) GOTOF DESTINO
```

ou

```
IF NOT R10 GOTOB PARTIDA
```

NOT refere-se apenas a um operando.

Exemplo 3: Operadores lógicos por Bits

```
IF $MC_RESET_MODE_MASK B_AND 'B10000' GOTOF ACT_PLANE
```

1.5 Correção da precisão em erros de comparação (TRUNC)

Função

O comando TRUNC corta o operando multiplicado por um fator de precisão.

Precisão ajustável em comandos de comparação

Os dados de programa de peça do tipo REAL são representados internamente em formato IEEE com 64 Bits. Devido a esta forma de representação as casas decimais podem ser imprecisas, o que pode gerar resultados inesperados se compararmos aos valores calculados.

Igualdade relativa

Para que a precisão ressaltada através da forma de representação não falsifique o fluxo do programa, então com os comandos de comparação não é realizado o controle sobre a igualdade absoluta, e sim sobre a igualdade relativa.

Sintaxe

Correção da precisão em erros de comparação

```
TRUNC (R1*1000)
```

Significado

`TRUNC`: Corte das casas decimais

Igualdade relativa considerada de 10^{-12} com

- Igualdade: (==)
- Desigualdade: (<>)
- Maior-Igual: (>=)
- Menor-Igual: (<=)
- Maior/Menor: (><) com igualdade absoluta
- Maior: (>)
- Menor: (<)

Compatibilidade

Por motivos de compatibilidade o controle sobre a igualdade relativa pode ser desativada no caso do (>) e (<) através da definição do dado de máquina MD10280 $MN_ PROG_FUNCTION_MASK Bit0 = 1.

Indicação

As comparações com dados do tipo REAL são condenados com uma certa imprecisão devido aos motivos já conhecidos. No caso de desvios não aceitáveis deve-se desviar para o cálculo INTEGER, onde os operandos são multiplicados por um fator de precisão e depois cortados com TRUNC.

Ações sincronizadas

O comportamento descrito dos comandos de comparação também se aplica nas ações sincronizadas.

Exemplos

Exemplo 1: Considerações de precisão

```
Código de programa                                  Comentário
N40 R1=61.01 R2=61.02 R3=0.01                       ;  Atribuição dos valores iniciais
N41 IF ABS(R2-R1) > R3 GOTOF ERRO                   ;  O salto seria executado até então
N42 M30                                             ;  Fim do programa
N43 ERRO: SETAL(66000)                              ;
R1=61.01 R2=61.02 R3=0.01                           ;  Atribuição dos valores iniciais
R11=TRUNC(R1*1000) R12=TRUNC(R2*1000)               ;  Correção da precisão
R13=TRUNC(R3*1000)
IF ABS(R12-R11) > R13 GOTOF ERRO                    ;  O salto não será mais executado
M30                                                 ;  Fim do programa
ERRO: SETAL(66000)                                  ;
```

Exemplo 2: Formação e avaliação do quociente dos dois operandos

```
Código de programa                                  Comentário
R1=61.01 R2=61.02 R3=0.01                           ;  Atribuição dos valores iniciais
IF ABS((R2-R1)/R3)-1) > 10EX-5 GOTOF ERRO            ;  O salto não será executado
M30                                                 ;  Fim do programa
ERRO: SETAL(66000)                                  ;
```

1.6 Mínimo, máximo e área de variáveis (MINVAL, MAXVAL, BOUND)

Função

Com os comandos MINVAL e MAXVAL os valores de duas variáveis podem ser comparados entre si. Como resultado é retornado o menor valor (com MINVAL) ou o maior valor (com MAXVAL).

Com o comando BOUND pode ser realizado o controle se o valor de uma variável de controle está dentro da faixa de valores definida.

Sintaxe

```
<menor valor>=MINVAL(<variável1>,<variável2>)
<maior valor>=MAXVAL(<variável1>,<variável2>)
<valor retornado>=<BOUND>(<mínimo>,<máximo>,<variável de controle>)
```

Significado

MINVAL:	Determina o **menor** valor de duas variáveis (<variável1>, <variável2>)
<menor valor>:	Variável de resultado para o comando MINVAL É passado para o menor valor de variável.
MAXVAL:	Determina o **maior** valor de duas variáveis (<variável1>, <variável2>)
<maior valor>:	Variável de resultado para o comando MAXVAL É passado para o maior valor de variável.
BOUND:	Controla se a variável (<variável de controle>) está dentro de uma faixa de valores definida.
<mínimo>:	Variável que é definida com o valor mínimo da faixa de valores
<máximo>:	Variável que é definida com o valor máximo da faixa de valores
<valor de retorno>:	Variável de resultado para o comando BOUND Quando o valor da variável de controle estiver dentro da faixa de valores definida, então a variável de resultado é passada para o valor da variável de controle. Quando o valor da variável de controle for maior que o valor máximo, então a variável de resultado é passada para o valor máximo da faixa de definição. Quando o valor da variável de controle for menor que o valor mínimo, então a variável de resultado é passada para o valor mínimo da faixa de definição.

Indicação

MINVAL, MAXVAL e BOUND também podem ser programadas em ações síncronas.

Indicação

Comportamento na igualdade

Em caso de igualdade, com `MINVAL/MAXVAL` é fornecido este mesmo valor. Com `BOUND` é retornado o valor da variável a ser controlada.

Exemplo

```
Código de programa                                  Comentário
DEF REAL rVar1=10.5, rVar2=33.7, rVar3, rVar4, rVar5, rValMin, rValMax, rRetVar
rValMin=MINVAL(rVar1,rVar2)                         ; rValMin é passado para o valor 10.5.
rValMax=MAXVAL(rVar1,rVar2)                         ; rValMax é passado para o valor 33.7.
rVar3=19.7
rRetVar=BOUND(rVar1,rVar2,rVar3)                    ; rVar3 está dentro dos limites, rRetVar é passado para
                                                      19.7.
rVar3=1.8
rRetVar=BOUND(rVar1,rVar2,rVar3)                    ; rVar3 está abaixo do limite mínimo, rRetVar é passado
                                                      para 10.5.

rVar3=45.2
rRetVar=BOUND(rVar1,rVar2,rVar3)                    ; rVar3 está acima do limite máximo, rRetVar é passado
                                                      para 33.7.
```

1.7 Prioridade das operações

Função

Cada operador é ordenado por um nível de prioridade. Na avaliação de uma expressão sempre são aplicados primeiro os operadores de nível de prioridade mais alto. No caso de operadores de mesma propriedade, a avaliação é realizada da esquerda para direita.

Em expressões aritméticas se pode definir a ordem de execução através de parênteses, e com isso, fugir das regras convencionais de prioridade.

Seqüência dos operadores

Do nível de prioridade mais alto até o mais baixo

1.	NOT, B_NOT	Negação, negação por Bits
2.	*, /, DIV, MOD	Multiplicação, divisão
3.	+, –	Adição, subtração
4.	B_AND	E bit a bit
5.	B_XOR	OU exclusivo bit a bit
6.	B_OR	OU bit a bit
7.	AND	E
8.	XOR	OU exclusivo
9.	OR	OU
10.	<<	Encadeamento de Strings, tipo de resultado STRING
11.	==, <>, >, <, >=, <=	Operadores de comparação

Indicação

O operador de encadeamento ":" para Frames não podem surgir com outros operadores em uma expressão. Por isso que uma categorização de prioridades para este operador não se faz necessária.

Exemplo de instrução If

```
If (otto==10) and (anna==20) gotof end
```

1.8 Possíveis conversões de tipo

Função

Conversão de tipos na atribuição

O valor numérico constante, a variável ou a expressão, que se atribui a uma variável, deve ser compatível com o tipo desta variável. Se esta condição for preenchida, então o tipo será convertido automaticamente durante a atribuição.

Possíveis conversões de tipos

para	REAL	INT	BOOL	CHAR	STRING	AXIS	FRAME
de							
REAL	sim	sim*	sim[1])	sim*	–	–	–
INT	sim	sim	sim[1])	sim[2])	–	–	–
BOOL	sim	sim	sim	sim	sim	–	–
CHAR	sim	sim	sim[1])	sim	sim	–	–
STRING	–	–	sim[4])	sim[3])	sim	–	–
AXIS	–	–	–	–	–	sim	–
FRAME	–	–	–	–	–	–	sim

Explanações

* Na conversão de tipos de REAL para INT um valor fracionado >= 0.5 é arredondado para cima, e em caso contrário, arredondado para baixo (veja a função ROUND).

1) O valor <> 0 corresponde à TRUE, o valor == 0 corresponde à FALSE

2) Se o valor estiver na faixa de números admissíveis

3) Se apenas 1 caractere

4) Tamanho de String 0 = >FALSE, caso contrário é TRUE

Indicação

Se durante a conversão um valor for maior que a faixa de destino, será emitida uma mensagem de erro.

Se aparecerem tipos mistos em uma expressão, então ser realiza automaticamente uma adaptação de tipos. As conversões de tipos também são possíveis em ações sincronizadas; veja o capítulo "Ações sincronizadas de movimentos, conversão implícita de tipos".

1.9 Operações de String

Operações de String

Além das operações clássicas "Atribuição" e "Comparação" também são possíveis as seguintes operações de String:

- Conversão de tipos para STRING (AXSTRING)
- Conversão de tipos de STRING (NUMBER, ISNUMBER, AXNAME)
- Encadeamento de Strings (<<)
- Mudança para letras minúsculas / maiúsculas (TOLOWER, TOUPPER)
- Definição do tamanho de uma String (STRLEN)
- Localização de caractere/String na String (INDEX, RINDEX, MINDEX, MATCH)
- Seleção de uma String parcial (SUBSTR)
- Seleção de um caractere individual (STRINGVAR, STRINGFELD)

Significado especial do caractere 0

O caractere 0 é interpretado internamente como identificação final de uma String. Se um caractere é substituído pelo caractere 0, então a String será encurtada com isso.

Exemplo:

```
Código de programa                              Comentário
DEF STRING[20] STRG="Eixo . está parado"
STRG[6]="X"
MSG(STRG)                                       ; Retorna a mensagem "Eixo X está
                                                  parado".
STRG[6]=0
MSG(STRG)                                       ; Retorna a mensagem "Eixo".
```

1.9.1 Conversão de tipos para STRING (AXSTRING)

Função

Através da função "Conversão de tipos para STRING" se pode utilizar variáveis de diferentes tipos como parte integrante de uma mensagem (MSG).

Ocorre na utilização do operador **<<** implicitamente para os tipos de dados INT, REAL, CHAR e BOOL (veja " Encadeamento de Strings (<<) (Página 78) ").

Um valor INT é convertido na forma normal e legível. Para valores REAL são indicadas até 10 casas decimais.

Com o comando `AXSTRING` se pode converter variáveis do tipo AXIS para o tipo STRING.

Sintaxe

```
<STRING_ERG> = << <ocup._tipo>
<STRING_ERG> = AXSTRING(<identificador de eixo>)
```

Significado

`<STRING_ERG>`:	Variável para o resultado da conversão de tipos Tipo: STRING
`<ocup._tipo>`:	Tipos de variáveis INT, REAL, CHAR, STRING e BOOL
`AXSTRING`:	O comando `AXSTRING` retorna o identificador de eixo indicado como String.
`<identificador de eixo>`:	Variável para identificador de eixo Tipo: AXIS

Indicação

As variáveis FRAME não podem ser convertidas.

Exemplos

Exemplo 1:

```
MSG("Posição:"<<$AA_IM[X])
```

Exemplo 2: AXSTRING

Código de programa	Comentário
`DEF STRING[32] STRING_ERG`	
`STRING_ERG=AXSTRING(X)`	`; STRING_ERG == "X"`

1.9.2 Conversão de tipos de STRING (NUMBER, ISNUMBER, AXNAME)

Função

Com o comando NUMBER se converte de STRING para REAL. A capacidade de conversão pode ser verificada através do comando ISNUMBER.

Com o comando AXNAME se converte uma String no tipo de dado AXIS.

Sintaxe

```
<REAL_ERG>=NUMBER("<String>")
<BOOL_ERG>=ISNUMBER("<String>")
<AXIS_ERG>=AXNAME("<String>")
```

Significado

NUMBER: O comando NUMBER retorna o número representado através da <String> como valor REAL.

<String>: Variável a ser convertida do tipo STRING

<REAL_ERG>: Variável para o resultado da conversão de tipos com NUMBER

Tipo: REAL

ISNUMBER: Com o comando ISNUMBER pode ser verificado se a <String> pode ser convertida em um número válido.

<BOOL_ERG>: Variável para o resultado da consulta com ISNUMBER

Tipo: BOOL

Valor: TRUE — ISNUMBER retorna o valor TRUE se a <String> apresenta um número REAL válido conforme as regras da linguagem.

FALSE — Se ISNUMBER retornar o valor FALSE, então na chamada do NUMBER é emitido o alarme com a mesma <String>.

AXNAME: O comando AXNAME converte a <String> especificada em um identificador de eixo.

Nota:
Se a <String> não pode ser associada a nenhum identificador de eixo configurado, então será emitido um alarme.

<AXIS_ERG>: Variável para o resultado da conversão de tipos com AXNAME

Tipo: AXIS

Exemplo

Código de programa	Comentário
DEF BOOL BOOL_ERG	
DEF REAL REAL_ERG	
DEF AXIS AXIS_ERG	
BOOL_ERG=ISNUMBER("1234.9876Ex-7")	; BOOL_ERG == TRUE
BOOL_ERG=ISNUMBER("1234XYZ")	; BOOL_ERG == FALSE
REAL_ERG=NUMBER("1234.9876Ex-7")	; REAL_ERG == 1234.9876Ex-7
AXIS_ERG=AXNAME("X")	; AXIS_ERG == X

1.9.3 Encadeamento de Strings (<<)

Função

A função "Encadeamento de Strings" oferece a capacidade de compor uma String a partir de partes individuais.

O encadeamento é realizado através do operador "**<<**". Este operador tem como tipo de destino STRING para todas combinações dos tipos básicos CHAR, BOOL, INT, REAL e STRING. Uma eventual conversão que for necessária será realizada conforme as regras existentes.

Sintaxe

```
<ocup._tipo> << <ocup._tipo>
```

Significado

`<ocup._tipo>`: Variável do tipo CHAR, BOOL, INT, REAL ou STRING

`<<` : Operador para o encadeamento de variáveis (<ocup._tipo) para uma seqüência de caracteres composta (tipo STRING).

Este operador também está disponível como uma variante "unária". Dessa forma é possível executar uma conversão explícita de tipos para STRING (não para FRAME e AXIS):

```
<< <ocup._tipo>
```

Por exemplo, se pode compor uma mensagem destas ou um comando a partir de listas de texto e inserir parâmetros (algo como um nome de módulo):

```
MSG(STRG_TAB[LOAD_IDX]<<NOME_MODULO)
```

CUIDADO

Os resultados intermediários no encadeamento de Strings não podem exceder o tamanho máximo de String.

Indicação

Os tipos FRAME e AXIS não podem ser utilizados juntos com o operador "<<".

Exemplos

Exemplo 1: Encadeamento de Strings

```
Código de programa                                  Comentário
DEF INT IDX=2
DEF REAL VALUE=9.654
DEF STRING[20] STRG="ÍNDICE:2"
IF STRG=="Índice:"<<IDX GOTOF NO_MSG
MSG("Índice:"<<IDX<<"/valor:"<<VALUE)               ; Exibição:
                                                      "Índice:2/valor:9.654"
NO_MSG:
```

Exemplo 2: Conversão explícita de tipos com <<

```
Código de programa          Comentário
DEF REAL VALUE=3.5
<<VALUE                     ; A variável especificada de tipo REAL é convertida para
                              o tipo STRING.
```

1.9.4 Mudança para letras minúsculas / letras maiúsculas (TOLOWER, TOUPPER)

Função

A função "Mudança para letras minúsculas / maiúsculas" permite a mudança de todas as letras de uma seqüência de caracteres para uma representação uniforme.

Sintaxe

```
<STRING_ERG>=TOUPPER("<String>")
<STRING_ERG>=TOLOWER("<String>")
```

Significado

TOUPPER:	Com o comando TOUPPER todas as letras de uma seqüência de caracteres muda para letras **maiúsculas**.
TOLOWER:	Com o comando TOLOWER todas as letras de uma seqüência de caracteres muda para letras **minúsculas**.
<String>:	Seqüência de caracteres que deve ser convertida Tipo: STRING
<STRING_ERG>:	Variável para o resultado da conversão Tipo: STRING

Exemplo

Visto que também é possível mostrar as especificações de usuário na interface de operação, a representação uniforme se obtém através de letras minúsculas ou maiúsculas:

Código de programa

```
DEF STRING [29] STRG
...
IF "LEARN.CNC"==TOUPPER(STRG) GOTOF LOAD_LEARN
```

1.9.5 Definir o tamanho de uma Strings (STRLEN)

Função

Com o comando STRLEN é possível definir o tamanho de uma seqüência de caracteres.

Sintaxe

```
<INT_ERG>=STRLEN("<STRING>")
```

Significado

STRLEN:	Com o comando STRLEN se define o tamanho da seqüência de caracteres especificada. É retornado o número de caracteres que são contados desde o início da seqüência de caracteres e que não são caracteres 0.
<String>:	Seqüência de caracteres cujo tamanho deve ser determinado Tipo: STRING
<INT_ERG>:	Variável para o resultado da definição Tipo: INT

Exemplo

A função junto com o acesso dos caracteres individuais permite determinar o fim de uma seqüência de caracteres:

Código de programa

```
IF (STRLEN(NOME_MODULO)>10) GOTOF ERRO
```

1.9.6 Localizar caractere/String na String (INDEX, RINDEX, MINDEX, MATCH)

Função

Esta funcionalidade permite localizar um único caractere ou uma String dentro de outra String. Os resultados de função indicam em qual posição da String foi encontrado o caractere / a String dentro da String analisada.

Sintaxe

`INT_ERG=INDEX(STRING,CHAR)` ; **Tipo de resultado: INT**

`INT_ERG=RINDEX(STRING,CHAR)` ; **Tipo de resultado: INT**

`INT_ERG=MINDEX(STRING,STRING)` ; **Tipo de resultado: INT**

`INT_ERG=MATCH(STRING,STRING)` ; **Tipo de resultado: INT**

Semântica

Funções de busca: Elas retornam a posição na String (primeiro parâmetro) onde a localização teve êxito. Se o caractere ou a String não puder ser localizado, então se retorna o valor -1. Neste caso o primeiro caractere possui a posição 0.

Significado

INDEX:	Busca (do começo) o caractere especificado como segundo parâmetro no primeiro parâmetro.
RINDEX:	Busca (do fim) o caractere especificado como segundo parâmetro no primeiro parâmetro.
MINDEX:	Corresponde à função INDEX, com exceção de que se transmite uma lista de caracteres (como String) de onde o índice do primeiro caractere encontrado é retornado.
MATCH:	Busca uma String dentro de uma String.

Dessa forma as Strings podem ser desmembradas conforme determinados critérios, algo como posições com espaço vazio ou sinais separadores de caminhos / atalhos ("/").

Exemplo

Desmembramento de uma entrada em nomes de caminho e de módulo

```
Código de programa                                                  Comentário
DEF INT PFADIDX, PROGIDX
DEF STRING[26] ESPECIFICACAO
DEF INT LISTIDX
ESPECIFICACAO = "/_N_MPF_DIR/_N_EXECUTE_MPF"
LISTIDX = MINDEX (ESPECIFICACAO, "M,N,O,P") + 1                     ; Como valor no LISTIDX se retorna o 3;
                                                                      pois o "N" é o primeiro caractere no
                                                                      parâmetro ESPECIFICACAO da lista de
                                                                      seleção partindo-se do início.
PFADIDX = INDEX (ESPECIFICACAO, "/") +1                             ; Com isso se aplica: PFADIDX = 1
PROGIDX = RINDEX (ESPECIFICACAO, "/") +1                            ; Com isso se aplica: PROGIDX = 12
                                                                      Com a ajuda da função SUBSTR inserida
                                                                      na próxima secção se pode desmembrar
                                                                      a variável ESPECIFICACAO nos
                                                                      componentes ;"caminho";e "módulo":
VARIABLE = SUBSTR (ESPECIFICACAO, PFADIDX, PROGIDX-PFADIDX-1)       ; Retorna "_N_MPF_DIR"
VARIABLE = SUBSTR (ESPECIFICACAO, PROGIDX)                          ; Retorna "_N_EXECUTE_MPF"
```

1.9.7 Seleção de uma String parcial (SUBSTR)

Função

Esta funcionalidade permite localizar uma String parcial de dentro de uma String. Para isso se especifica o índice do primeiro caractere e o tamanho desejado. Se não for especificada a informação do tamanho, espera-se a String restante.

Sintaxe

`STRING_ERG = SUBSTR (STRING,INT)` ; **Tipo de resultado: INT**

`STRING_ERG = SUBSTR(STRING,INT, INT)` ; **Tipo de resultado: INT**

Semântica

No primeiro caso é retornada a String parcial desde a posição definida pelo segundo parâmetro até o fim da String.

No segundo caso a String resultante é limitada no tamanho máximo dado pelo terceiro parâmetro.

Se a posição inicial estiver após o fim da String, será retornada a String vazia (" ").

Se a posição inicial ou o tamanho for negativo, será emitido um alarme.

Exemplo

Código de programa	**Comentário**
`DEF STRING[29] ERG`	
`ERG = SUBSTR ("CONFIRMACAO:10 a 99", 10, 2)`	`; Com isso se aplica: ERG == "10"`

1.9.8 Seleção de um caractere individual (STRINGVAR, STRINGFELD)

Função

Esta funcionalidade permite a seleção de caracteres individuais de uma String. Isto afeta tanto o acesso de leitura como o acesso de gravação.

Sintaxe

CHAR_ERG = STRINGVAR [IDX] ; **Tipo de resultado: CHAR**

CHAR_ERG = STRINGFELD [IDX_FELD, IDX_CHAR] ; **Tipo de resultado: CHAR**

Semântica

Se realiza a leitura/gravação do caractere dentro da String que estiver na posição especificada. Se a indicação de posição for negativa ou então maior que a String, será emitido um alarme.

Exemplo de mensagens:

Emprego de um identificador de eixo em uma String composta previamente.

```
DEF STRING [50] MENSAGEM = "O eixo n alcançou a posição"
MENSAGEM [6] = "X"
MSG (MENSAGEM)                                    ; Retorna a mensagem "O eixo X
                                                    alcançou a posição"
```

Parâmetro

O acesso a caracteres individuais somente é possível em variáveis definidas pelo usuário (dados LUD, GUD e PUD).

Além disso, esse tipo de acesso é possível apenas em uma chamada de subrotina para parâmetros do tipo "Call-By-Value".

Exemplos

Exemplo 1: Acesso de caracteres individuais em um dado de sistema, dado de máquina, …

```
DEF STRING [50] STRG
DEF CHAR CONFIRMACAO
…
STRG = $P_MMCA
CONFIRMACAO = STRG [0]                            ;  Avaliação do componente de
                                                     confirmação
```

Exemplo 2: Acesso de caractere individual com parâmetro Call-By-Reference

```
DEF STRING [50] STRG
DEF CHAR CHR1
EXTERN UP_CALL (VAR CHAR1)                        ;  Parâmetro Call-By-Reference!
…
CHR1 = STRG [5]
UP_CALL (CHR1)                                    ;  Call-By-Reference
STRG [5] = CHR1
```

1.10 Saltos e ramificações de programa

1.10.1 Salto de retorno ao início do programa (GOTOS)

Função

Com o comando GOTOS é possível saltar retornando ao início de um programa principal ou de uma subrotina para repetição do programa.

Através de dados de máquina se pode ajustar para que em cada salto de retorno ao início do programa:

- o tempo de execução do programa passe novamente para "0" (seja zerado).
- a contagem de peças seja incrementada com o valor "1".

Sintaxe

```
GOTOS
```

Significado

GOTOS: Instrução de salto com destino de salto no início do programa.
A execução é controlada através do sinal de interface NC/PLC:
DB21, ... DBX384.0 (controle da bifurcação do programa)

Valor:	Significado:
0	Nenhum salto de retorno ao início do programa. A execução do programa é continuada com o próximo bloco de programa de peça após o GOTOS.
1	Salto de retorno ao início do programa. Se repete o programa de peça.

Condições gerais

- GOTOS inicia internamente um STOPRE (parada de pré-processamento).
- Em um programa de peça com definições de dados (variáveis LUD) com o GOTOS se realiza o salto para o primeiro bloco de programa após a parte das definições, isto é, as definições não são executadas novamente. Por isso que as variáveis definidas preservam o valor obtido no bloco do GOTOS e não são resetadas nos valores padrão programados na parte das definições.
- O comando GOTOS não está disponível em ações sincronizadas e em ciclos tecnológicos.

Exemplo

```
Código de programa      Comentário
N10 ...                 ; Início do programa.
...
N90 GOTOS               ; Salto até o início do programa.
...
```

1.10.2 Saltos de programa até marcadores de salto (GOTOB, GOTOF, GOTO, GOTOC)

Função

Em um programa podem ser colocados marcadores de salto (Labels) de onde se pode saltar para outros pontos dentro do mesmo programa através dos comandos `GOTOF`, `GOTOB`, `GOTO` e `GOTOC`. A execução do programa é continuada com a instrução que seguir imediatamente após o marcador de salto. Com isso as bifurcações são realizáveis dentro do programa.

Além dos marcadores de salto também é possível o uso de números de bloco principal e secundário como destinos de salto.

Se estiver formulada a condição de salto (`IF ...`) antes da instrução de salto, então o salto de programa somente será realizado quando a condição de salto for preenchida.

Sintaxe

```
GOTOB <destino de salto>
IF <condição de salto> = TRUE GOTOB <destino de salto>
GOTOF <destino de salto>
IF <condição de salto> = TRUE GOTOF <destino de salto>
GOTO <destino de salto>
IF <condição de salto> = TRUE GOTO <destino de salto>
GOTOC <destino de salto>
IF <condição de salto> = TRUE GOTOC <destino de salto>
```

Significado

`GOTOB`:	Instrução de salto com destino de salto em direção ao início do programa.
`GOTOF`:	Instrução de salto com destino de salto em direção ao fim do programa.
`GOTO`:	Instrução de salto com localização do destino de salto. A localização é realizada primeiro em direção ao fim do programa, depois em direção ao início do programa.
`GOTOC`:	Efeito como o GOTO, mas com a diferença que o alarme 14080 "Destino de salto não encontrado" é omitido. Isto significa que a execução do programa não é interrompida em um caso de localização de destino de salto sem resultado, mas através do comando GOTOC se prossegue com a execução da próxima linha do programa.
`<destino de salto>`:	Parâmetro do destino de salto As possíveis indicações são: <marcador de salto>: O destino de salto é o marcador de salto colocado no programa com nome definido pelo usuário: `<marcador de salto>:`

	<número de bloco>:	O destino de salto é um número de bloco principal ou secundário (p. ex.: `200`, `N300`)
	Variável do tipo STRING:	Destino de salto variável. A variável serve para um marcador de salto ou um número de bloco.
`IF`:		Palavra-chave para formulação da condição de salto. A condição de salto permite todas operações de comparação e operações lógicas (resultado: TRUE ou FALSE). O salto de programa é executado quando o resultado desta operação for TRUE.

Indicação

Marcadores de salto (Labels)

Os marcadores de salto sempre estão no início de um bloco. Se um número de programa estiver presente, o marcador de salto está imediatamente após o número do bloco.

Para a atribuição de nomes de marcadores de salto são aplicadas as seguintes regras:

- Número de caracteres:
  - pelo menos 2
  - no máximo 32
- Os caracteres permitidos são:
  - Letras
  - Números
  - Sublinhados
- Os primeiros dois caracteres devem ser letras ou sublinhados.
- Após o nome do marcador de salto segue um duplo ponto (":").

Condições gerais

- O destino de salto somente pode ser um bloco com marcador de salto ou número de bloco que está **dentro** do programa.
- Uma instrução de salto sem condição de salto deve ser programada em um bloco separado. Esta restrição não se aplica em instruções de salto com condição de salto. Aqui podem ser formuladas várias instruções de salto em um mesmo bloco.
- Nos programas com instruções de salto sem condições de salto não é obrigatório que o fim de programa `M2`/`M30` esteja no fim do programa.

Exemplos

Exemplo 1: Saltos para marcadores de salto

```
Código de programa                Comentário
N10 …
N20 GOTOF Label_1                 ; Salto em direção ao fim do programa até o
                                    marcador de salto "Label_1".
N30 …
N40 Label_0: R1=R2+R3             ; Marcador de salto "Label_0" definido.
N50 …
N60 Label_1:                      ; Marcador de salto "Label_1" definido.
N70 …
N80 GOTOB Label_0                 ; Salto em direção ao início do programa até o
                                    marcador de salto "Label_0".
N90 …
```

Exemplo 2: Salto indireto até o número do bloco

```
Código de programa                Comentário
N5 R10=100
N10 GOTOF "N"<<R10                ; Salto até o bloco cujo número de bloco está no
                                    R10.
...
N90 ...
N100 ...                          ; Destino de salto
N110 ...
...
```

Exemplo 3: Salto até o destino de salto variável

```
Código de programa                Comentário
DEF STRING[20] DESTINO
DESTINO = "Marcador2"
GOTOF DESTINO                     ; Salto em direção ao fim do programa até o destino
                                    de salto variável DESTINO.
Marcador1: T="Broca1"
...
Marcador2: T="Broca2"             ; Destino de salto
...
```

Exemplo 4: Salto com condição de salto

Código de programa	Comentário
N40 R1=30 R2=60 R3=10 R4=11 R5=50 R6=20	; Atribuição dos valores iniciais.
N41 LA1: G0 X=R2*COS(R1)+R5 Y=R2*SIN(R1)+R6	; Marcador de salto LA1 definido.
N42 R1=R1+R3 R4=R4-1	
N43 IF R4>0 GOTOB LA1	; Quando a condição de salto estiver preenchida, então ocorre o salto em direção ao início do programa até o marcador de salto LA1.
N44 M30	; Fim do programa

1.10.3 Bifurcação do programa (CASE ... OF ... DEFAULT ...)

Função

A função CASE oferece a opção de controlar o atual valor (tipo: INT) de uma variável ou uma função de cálculo e, em função do resultado, saltar para diferentes pontos no programa.

Sintaxe

```
CASE(<expressão>) OF <constante_1> GOTOF <destino_salto_1>
<constante_2> GOTOF <destino_salto_2> ... DEFAULT GOTOF
<destino_salto_n>
```

Significado

CASE:	Instrução de salto
<expressão>:	Variável ou função de cálculo
OF:	Palavra-chave para formulação das bifurcações condicionais de programa
<constante_1>:	Primeiro valor constante especificado para a variável ou função de cálculo Tipo: INT
<constante_2>:	Segundo valor constante especificado para a variável ou função de cálculo Tipo: INT
DEFAULT:	Para os casos em que a variável ou a função de cálculo não aceitar nenhum dos valores constantes especificados, se pode definir um destino de salto com a instrução DEFAULT. **Nota:** Se a instrução DEFAULT não for programada, então nestes casos o bloco seguinte da instrução CASE passará a ser o destino de salto.

GOTOF:	Instrução de salto com destino de salto em direção ao fim do programa. Ao invés do GOTOF também podem ser programados todos os demais comandos GOTO (veja o assunto "Saltos de programa até os marcadores de salto").
<destino_salto_1>:	A bifurcação vai para este destino de salto se o valor da variável ou da função de cálculo corresponder à primeira constante especificada. O destino de salto pode ser especificado da seguinte forma: **<marcador de salto>**: O destino de salto é o marcador de salto colocado no programa com nome definido pelo usuário: <marcador de salto>: **<número de bloco>**: O destino de salto é um número de bloco principal ou secundário (p. ex.: 200, N300) Variável do tipo STRING: Destino de salto variável. A variável serve para um marcador de salto ou um número de bloco.
<destino_salto_2>:	A bifurcação vai para este destino de salto se o valor da variável ou da função de cálculo corresponder à segunda constante especificada.
<destino_salto_n>:	A bifurcação vai para este destino de salto se o valor da variável não corresponder a nenhum dos valores constantes especificados.

Exemplo

Código de programa

```
...
N20 DEF INT VAR1 VAR2 VAR3
N30 CASE(VAR1+VAR2-VAR3) OF 7 GOTOF Label_1 9 GOTOF Label_2 DEFAULT GOTOF Label_3
N40 Label_1: G0 X1 Y1
N50 Label_2: G0 X2 Y2
N60 Label_3: G0 X3 Y3
...
```

A instrução CASE do N30 define as seguintes possibilidades de bifurcação de programa:

1. Se o valor da função de cálculo VAR1+VAR2-VAR3 = 7, então se salta até o bloco com a definição de marcador de salto "Label_1" (→ N40).
2. Se o valor da função de cálculo VAR1+VAR2-VAR3 = 9, então se salta até o bloco com a definição de marcador de salto "Label_2" (→ N50).
3. Se o valor da função de cálculo VAR1+VAR2-VAR3 não for 7 nem 9, então se salta até o bloco com a definição de marcador de salto "Label_3" (→ N60).

1.11 Repetição de partes do programa (REPEAT, REPEATB, ENDLABEL, P)

Função

A função de repetição de partes do programa possibilita a repetição de partes do programa já escritas dentro do programa em qualquer tipo de ordem ou formação.

As linhas e áreas de programa que devem ser repetidas precisam ser marcadas com marcadores de salto (Labels).

Indicação

Marcadores de salto (Labels)

Os marcadores de salto sempre estão no início de um bloco. Se um número de programa estiver presente, o marcador de salto está imediatamente após o número do bloco.

Para a atribuição de nomes de marcadores de salto são aplicadas as seguintes regras:

- Número de caracteres:
  - pelo menos 2
  - no máximo 32
- Os caracteres permitidos são:
  - Letras
  - Números
  - Sublinhados
- Os primeiros dois caracteres devem ser letras ou sublinhados.
- Após o nome do marcador de salto segue um duplo ponto (":").

Sintaxe

1. Repetição de linhas de programa individuais:

```
<marcador de salto>: ...
...
REPEATB <marcador de salto> P=<n>
...
```

2. Repetição de área de programa entre marcador de salto e a instrução REPEAT:

```
<marcador de salto>: ...
...
REPEAT <marcador de salto> P=<n>
...
```

3. Repetição de área entre dois marcadores de salto:

```
<marcador de salto inicial>: ...
...
<marcador de salto final>: ...
...
REPEAT <marcador de salto inicial> <marcador de salto final> P=<n>
...
```

Indicação

Não é possível colocar a instrução `REPEAT` com os dois marcadores de salto entre colchetes/parênteses. Se o `<marcador de salto inicial>` for encontrado antes da instrução `REPEAT` e o `<marcador de salto final>` não for alcançado antes da instrução `REPEAT`, então é executada a repetição entre o `<marcador de salto inicial>` e a instrução `REPEAT`.

4. Repetição entre o marcador de salto e o ENDLABEL:

```
<marcador de salto>: ...
...
ENDLABEL: ...
...
REPEAT <marcador de salto> P=<n>
...
```

Indicação

Não é possível colocar a instrução `REPEAT` com o `<marcador de salto>` e o `ENDLABEL` entre colchetes/parênteses. Se o `<marcador de salto>` for encontrado antes da instrução `REPEAT` e o `ENDLABEL` não for alcançado antes da instrução `REPEAT`, então é executada a repetição entre o `<marcador de salto>` e a instrução `REPEAT`.

Significado

REPEATB:	Comando para repetição de uma linha de programa
REPEAT:	Comando para repetição de uma área de programa
<marcador de salto>:	O <marcador de salto> identifica: • a linha de programa que deve ser repetida (com REPEATB) ou • o início da área de programa que deve ser repetida (com REPEAT) A linha de programa marcada com o <marcador de salto> pode estar antes ou depois da instrução REPEAT/REPEATB. A localização é realizada primeiramente em direção ao início do programa. Se o marcador de salto não for encontrado nesta direção, então a localização será realizada em direção ao fim do programa. **Exceção:** Se a área de programa entre o marcador de salto e a instrução REPEAT deve ser repetida (veja item 2. em Sintaxe), então a linha de programa marcada com o <marcador de salto> deve estar **antes** da instrução REPEAT, pois neste caso a localização **somente** é realizada em direção ao início do programa. Se a linha com o <marcador de salto> contém outras instruções, então estas serão executadas em toda repetição.
ENDLABEL:	Palavra-chave que marca o fim de uma área de programa que deve ser repetida Se a linha com o ENDLABEL contém outras instruções, então estas serão executadas em toda repetição. O ENDLABEL pode ser utilizado várias vezes no programa.
P:	Endereço para especificação da quantidade de repetições
<n>:	Quantidade de repetições da parte do programa Tipo: INT A parte do programa a ser repetida é repetida <n> vezes. Após a última repetição, o programa é continuado com a linha seguinte à linha da instrução REPEAT/REPEATB. **Nota:** Se nenhum P=<n> for especificado, então a parte do programa a ser repetida será repetida apenas uma vez.

Exemplos

Exemplo 1: Repetição de linha de programa individual

```
Código de programa                    Comentário
N10 POSITION1: X10 Y20
N20 POSITION2: CYCLE(0,,9,8)          ; Ciclo de posição
N30 ...
N40 REPEATB POSITION1 P=5             ; Execute cinco vezes o BLOCO N10.
N50 REPEATB POSITION2                 ; Execute uma vez o bloco N20.
N60 ...
N70 M30
```

Exemplo 2: Repetição de área de programa entre marcador de salto e a instrução REPEAT

```
Código de programa          Comentário
N5 R10=15
N10 Begin: R10=R10+1        ; Largura
N20 Z=10-R10
N30 G1 X=R10 F200
N40 Y=R10
N50 X=-R10
N60 Y=-R10
N70 Z=10+R10
N80 REPEAT BEGIN P=4        ; Execute quatro vezes a área do N10 ao N70.
N90 Z10
N100 M30
```

Exemplo 3: Repetição da área entre dois marcadores de salto

```
Código de programa              Comentário
N5 R10=15
N10 Begin: R10=R10+1            ; Largura
N20 Z=10-R10
N30 G1 X=R10 F200
N40 Y=R10
N50 X=-R10
N60 Y=-R10
N70 END: Z=10
N80 Z10
N90 CYCLE(10,20,30)
N100 REPEAT BEGIN END P=3       ; Execute três vezes a área do N10 ao N70.
N110 Z10
N120 M30
```

Exemplo 4: Repetição entre o marcador de salto e o ENDLABEL

```
Código de programa          Comentário
N10 G1 F300 Z-10
N20 BEGIN1:
N30 X10
N40 Y10
N50 BEGIN2:
N60 X20
N70 Y30
N80 ENDLABEL: Z10
N90 X0 Y0 Z0
N100 Z-10
N110 BEGIN3: X20
N120 Y30
N130 REPEAT BEGIN3 P=3      ; Execute três vezes a área do N110 ao N120.
N140 REPEAT BEGIN2 P=2      ; Execute duas vezes a área do N50 ao N80.
N150 M100
N160 REPEAT BEGIN1 P=2      ; Execute duas vezes a área do N20 ao N80.
N170 Z10
N180 X0 Y0
N190 M30
```

Exemplo 5: Execução de operação de fresamento, posição de furação com diferentes tecnologias

```
Código de programa          Comentário
N10 BROCACENTRAGEM()        ; Carregar a broca de centragem.
N20 POS_1:                  ; Posições de furação 1
N30 X1 Y1
N40 X2
N50 Y2
N60 X3 Y3
N70 ENDLABEL:
N80 POS_2:                  ; Posições de furação 2
N90 X10 Y5
N100 X9 Y-5
N110 X3 Y3
N120 ENDLABEL:
N130 BROCA()                ; Carregar a broca e ciclo de furação.
N140 ROSCA(6)               ; Carregar o macho M6 e ciclo de rosqueamento.
N150 REPEAT POS_1           ; Repita uma vez o segmento de programa a partir do POS_1
                              até o ENDLABEL.
```

```
Código de programa          Comentário
N160 BROCA()                ; Carregar a broca e ciclo de furação.
N170 ROSCA(8)               ; Carregar o macho M8 e ciclo de rosqueamento.
N180 REPEAT POS_2           ; Repita uma vez o segmento de programa a partir do POS_2
                              até o ENDLABEL.
N190 M30
```

Outras informações

- A repetição de partes do programa pode ser chamada de forma aninhada. Cada chamada ocupa um nível de subrotina.
- Se durante a usinagem for programada uma repetição de uma parte do programa `M17` ou um `RET`, então a repetição da parte do programa será cancelada. O programa é continuado no bloco após a linha do `REPEAT`.
- Na atual exibição do programa se indica a repetição da parte do programa como um nível próprio de subrotina.
- Se durante a execução da parte do programa for ativada uma interrupção de nível, então a execução do programa continua após a chamada da execução da parte do programa.

  Exemplo:

```
Código de programa              Comentário
N5 R10=15
N10 BEGIN: R10=R10+1            ; Largura
N20 Z=10-R10
N30 G1 X=R10 F200
N40 Y=R10                       ; Interrupção de nível
N50 X=-R10
N60 Y=-R10
N70 END: Z10
N80 Z10
N90 CYCLE(10,20,30)
N100 REPEAT BEGIN END P=3
N120 Z10                        ; Continuar com a execução do programa.
N130 M30
```

- É possível combinar estruturas de controle e repetições de partes de programa. Entretanto, não deve haver nenhuma sobreposição. Uma repetição de partes de programa deve estar situada em uma bifurcação de estrutura de controle ou uma estrutura de controle deve estar situada em uma repetição de partes de programa.
- Se for realizada a mistura de saltos e repetições de partes de programa, os blocos são executados de forma seqüencial pura. Por exemplo, se um salto for realizado a partir de uma repetição de uma parte do programa, então a execução será realizada até ser encontrado o fim programado desta parte do programa.

Exemplo:

Código de programa

```
N10 G1 F300 Z-10
N20 BEGIN1:
N30 X=10
N40 Y=10
N50 GOTOF BEGIN2
N60 ENDLABEL:
N70 BEGIN2:
N80 X20
N90 Y30
N100 ENDLABEL: Z10
N110 X0 Y0 Z0
N120 Z-10
N130 REPEAT BEGIN1 P=2
N140 Z10
N150 X0 Y0
N160 M30
```

Indicação

A instrução `REPEAT` deve estar situada após os blocos de deslocamento.

1.12 Estruturas de controle

Função

Como padrão, o comando numérico executa os blocos NC na seqüência programada.

Esta seqüência pode ser variada mediante programação de blocos de programa alternativos e loop's de programa. A programação destas estruturas de controle é realizada com os elementos de estruturas de controle (palavras-chave) `IF...ELSE`, `LOOP`, `FOR`, `WHILE` e `REPEAT`.

⚠ CUIDADO

As estruturas de controle somente são possíveis dentro da parte de instrução de um programa. As definições no cabeçalho não podem ser executadas de forma condicional ou de forma repetida.

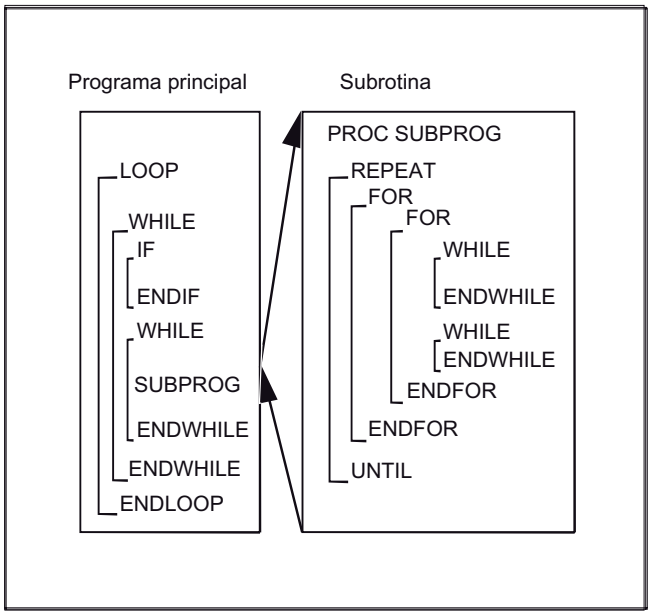
Da mesma forma, as palavras-chave para estruturas de controle não podem ser sobrepostas com macros. Não é realizado nenhum controle na definição de macros.

Efeito

As estruturas de controle são aplicadas localmente no programa.

Nível de aninhamento

Dentro de cada nível de subrotina é possível alcançar um nível de aninhamento de até 16 estruturas de controle.



Relação do tempo de execução

Com o modo de interpretação padrão ativado, mediante uso de saltos de programa, se pode alcançar uma execução mais rápida do que com estruturas de controle.

Nos ciclos pré-compilados não existe nenhuma diferença entre saltos de programa e estruturas de controle.

Condições gerais

- Os blocos com elementos de estrutura de controle não podem ser omitidos.
- Os marcadores de salto (Labels) não são permitidos em blocos com elementos de estrutura de controle.
- As estruturas de controle são executadas de modo interpretativo. Ao identificar o fim de um loop, sob consideração das estruturas de controle encontradas, será realizada uma busca pelo início do loop. Por isso que no modo de interpretação não se controla completamente a estrutura de bloco de um programa.
- Sempre se recomenda evitar o uso misto de estruturas de controle e saltos de programa.
- No pré-processamento de ciclos pode ser realizado o controle quanto ao aninhamento correto de estruturas de controle.

1.12.1 Loop de programa com alternativa (IF, ELSE, ENDIF)

Função

Uma construção com `IF` e `ELSE` é utilizada quando o loop de programa deve conter um bloco de programa alternativo: Se a condição `IF` estiver preenchida, então se executa o bloco de programa após o `IF`. Se a condição `IF`**não** estiver preenchida, então se executa o bloco de programa após o `ELSE`.

Indicação

Se nenhuma alternativa for necessária, então também se pode programar um loop IF sem a instrução `ELSE` e o bloco de programa após o `ELSE`.

Sintaxe

```
IF <condição>
...
ELSE
...
ENDIF
```

Significado

`IF`:	Inicia o loop IF.
`ELSE`:	Inicia o bloco de programa alternativo.
`ENDIF`:	Marca o fim do loop IF e realiza o salto de retorno ao início do loop.
`<condição>`:	Condição que decida qual bloco de programa deve ser executado.

Exemplo

Subrotina de troca de ferramentas

```
Código de programa                                  Comentário
PROC L6                                             ; Rotina de troca de ferramentas
N500 DEF INT TNR_ATUAL                              ; Variável para número T ativo
N510 DEF INT TNR_PRESELECAO                         ; Variável para número T pré-
                                                      selecionado
                                                    ; Determinar a ferramenta atual
N520 STOPRE
N530 IF $P_ISTEST                                   ; No modo de teste de programa
                                                      se ...
N540 TNR_ATUAL = $P_TOOLNO                          ; ... realiza a leitura da
                                                      "atual" ferramenta a partir do
                                                      contexto do programa.
N550 ELSE                                           ; Caso contrário se ...
N560 TNR_ATUAL = $TC_MPP6[9998,1]                   ; ... lê a ferramenta do fuso.
N570 ENDIF
N580 GETSELT(TNR_PRESELECAO)                        ; Leitura do número T da
                                                      ferramenta pré-selecionada no
                                                      fuso.
N590 IF TNR_ATUAL <> TNR_PRESELECAO                 ; Se a ferramenta pré-
                                                      selecionada ainda não for a
                                                      ferramenta atual, então se
                                                      deve ...
N600 G0 G40 G60 G90 SUPA X450 Y300 Z300 D0           ; ... Aproximar ponto de troca
                                                      de ferramentas ...
N610 M206                                           ; ... e executar a troca de
                                                      ferramenta.
N620 ENDIF
N630 M17
```

1.12.2 Loop de programa infinito (LOOP, ENDLOOP)

Função

O loop infinito se aplica em programas de processamento infinito. No fim do loop sempre se executa novamente o salto de retorno ao início do loop.

Sintaxe

```
LOOP
...
ENDLOOP
```

Significado

`LOOP`:	Inicia o loop infinito.
`ENDLOOP`:	Marca o fim do loop e realiza o salto de retorno ao início do loop.

Exemplo

Código de programa

```
...
LOOP
MSG("nenhum corte de ferramenta ativo")
M0
STOPRE
ENDLOOP
...
```

1.12.3 Loop contador (FOR ... TO ..., ENDFOR)

Função

O loop contador se utiliza quando uma execução de trabalho deve ser repetida com um número fixo de execuções.

Sintaxe

```
FOR <variável> = <valor inicial> TO <valor final>
...
ENDFOR
```

Significado

FOR:	Inicia o loop contador.
ENDFOR:	Marca o fim do loop e realiza o salto de retorno ao início do loop, enquanto o valor final da contagem não for alcançado.
<variável>:	Variável de contagem que é incrementada do valor inicial ao valor final pelo valor "1" em cada execução realizada. Tipo INT ou REAL **Nota:** Por exemplo, o tipo REAL é adotado quando são programados parâmetros R para um loop contador. Se a variável de contagem for do tipo REAL, seu valor será arredondado para um valor de número inteiro.
<valor inicial>:	Valor inicial da contagem Condição: O valor inicial deve ser menor que o valor final.
<valor final>:	Valor final da contagem

Exemplos

Exemplo 1: Variável INTEGER ou parâmetro R como variável de contagem

Variável INTEGER como variável de contagem:

```
Código de programa                  Comentário
DEF INT iVARIABLE1
R10=R12-R20*R1 R11=6
FOR iVARIABLE1= R10 TO R11          ; Variável de contagem = variável INTEGER
 R20=R21*R22+R33
ENDFOR
M30
```

Parâmetro R como variável de contagem:

```
Código de programa                    Comentário
R11=6
FOR R10=R12-R20*R1 TO R11             ; Variável de contagem = parâmetro R (variável
                                        Real)
 R20=R21*R22+R33
ENDFOR
M30
```

Exemplo 2: Produção de uma quantidade fixa de peças

```
Código de programa                    Comentário
DEF INT QTDEPECAS                     ; Define a variável do tipo INT com o nome
                                        "QTDEPECAS".
FOR QTDEPECAS = 0 TO 100              ; Inicia o loop contador. A variável "QTDEPECAS"
                                        é incrementada do valor inicial "0" até o valor
                                        final "100".
G01 …
ENDFOR                                ; Fim do loop contador.
M30
```

1.12.4 Loop de programa com condição no início do loop (WHILE, ENDWHILE)

Função

Em um loop WHILE a condição está situada no início do loop. O loop WHILE será executado enquanto a condição estiver preenchida.

Sintaxe

```
WHILE <condição>
...
ENDWHILE
```

Significado

`WHILE`:	Inicia o loop de programa.
`ENDWHILE`:	Marca o fim do loop e realiza o salto de retorno ao início do loop.
`<condição>`:	Condição que deve ser preenchida para que o loop WHILE seja executado.

Exemplo

```
Código de programa                          Comentário
...
WHILE $AA_IW[EIXOFURACAO] > -10             ; Chamada do loop WHILE sob a seguinte
                                              condição: o atual valor nominal WCS do
                                              eixo de furação deve ser maior que -10.
G1 G91 F250 AX[EIXOFURACAO] = -1
ENDWHILE
...
```

1.12.5 Loop de programa com condição no fim do loop (REPEAT, UNTIL)

Função

Em um loop REPEAT a condição está situada no fim do loop. O loop REPEAT é executado uma vez e será repetido enquanto a condição for preenchida.

Sintaxe

```
REPEAT
...
UNTIL <condição>
```

Significado

`REPEAT`:	Inicia o loop de programa.
`UNTIL`:	Marca o fim do loop e realiza o salto de retorno ao início do loop.
`<condição>`:	Condição que deve ser preenchida para que o loop REPEAT não seja mais executado.

Exemplo

```
Código de programa          Comentário
...
REPEAT                      ; Chamada do loop REPEAT.
...
UNTIL ...                   ; Controle para saber se a condição está preenchida.
...
```

1.12.6 Exemplo de programa com estruturas de controle aninhadas

Código de programa	**Comentário**
LOOP	
IF NOT $P_SEARCH	; sem localização de blocos
G01 G90 X0 Z10 F1000	
WHILE $AA_IM[X] <= 100	
G1 G91 X10 F500	; Modelo de furação
Z-F100	
Z5	
ENDWHILE	
Z10	
ELSE	
MSG("Na localização não se executa a furação")	
ENDIF	
$A_OUT[1] = 1	; próxima placa de furação
G4 F2	
ENDLOOP	
M30	

1.13 Coordenação de programa (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM)

Função

Canais

Um canal pode executar seu próprio programa, independentemente dos outros canais. Com isso o programa pode controlar os eixos e fusos temporariamente atribuídos.

Na colocação em funcionamento podem ser ajustados dois ou mais canais para o comando.

Coordenação de programa

Se forem utilizados vários canais para a usinagem de uma peça, então pode ser necessária uma sincronização dos processos de execução do programa.

Para essa coordenação do programa existem instruções (comandos) especiais. Cada uma está armazenada em um bloco.

Indicação

A coordenação de programa também é possível em um canal próprio.

Instruções para a coordenação do programa

- Especificação com indicação absoluta de caminho

	Neste caso o caminho absoluto é formado a partir das seguintes regras:
INIT (n,"/_HUGO_DIR/_N_nome_MPF") ou	- atual Directory/_N_nome_MPF "atual diretório" especifica o diretório selecionado da peça ou o diretório padrão /_N_MPF_DIR.
INIT (n,"/_N_MPF_DIR/_N_nome_MPF")	- Seleção de um determinado programa para execução em um determinado canal: n: Número do canal, valor conforme - configuração do comando Nome de programa completo
Exemplo:	**até o SW3:**
INIT(2,"/_N_WCS_DIR/_DRESSAGEM_MPF") G01F0.1 START INIT (2,"/_N_WCS_DIR/_N_INFERIOR_1_SPF")	Entre um comando **init** (sem sincronização) e um **NC-Start** precisa existir pelo menos um bloco executável. Para chamadas de subrotinas se deve adicionar o "_SPF" ao caminho.

- Especificação com a indicação relativa de caminho

Exemplo:	No caminho relativo se aplicam as mesmas regras das chamadas de subrotinas.
INIT(2,"DRESSAGEM")	
INIT(3,"INFERIOR_1_SPF")	Nas chamadas de subrotinas se devem adicionar o "_SPF" aos nomes de programa.

Parâmetro

Para a troca de dados entre os programas se pode utilizar as variáveis disponíveis em todos canais (variáveis globais específicas NCK). Caso contrário a criação do programa é feita separadamente para cada canal.

`INIT(n, indicação do caminho, modo de confirmação)`	Instrução para execução em um canal. Seleção de um determinado programa com indicação absoluta ou relativa do caminho.
`START (n, n)`	Inicialização dos programas selecionados nos outros canais. n,n: Enumeração dos números de canal: Valor conforme a configuração do comando
`WAITM (marcador n°, n, n, ...)`	Colocação do marcador "Marcador n°" em um canal próprio. Finalizar o bloco anterior com parada exata. Espera pelo marcador com o mesmo "Marcador n°" nos canais especificados "n" (o canal próprio não precisa ser especificado). O marcador não é deletado após a sincronização. Simultaneamente podem ser colocados no máx. 10 marcadores por canal.
`WAITMC (marcador n°, n, n,`	Colocação do marcador "Marcador n°" em um canal próprio. A parada exata somente é iniciada se os outros canais ainda não alcançaram o marcador. Espera pelo marcador com o mesmo "Marcador n°" nos canais especificados "n" (o canal próprio não precisa ser especificado). Assim que o marcador "Marcador n°" for alcançado nos canais especificados, o processamento continua sem finalizar a parada exata.

`WAITE (n, n, ...)`	Espera pelo fim do programa dos canais especificados (não especificar o canal próprio). Exemplo: Programação de um tempo de espera após o comando Start.
	`N30 START(2)`
	`N31 G4 F0.01`
	`N40 WAITE(2)`

`SETM (marcador n°, marcador n°,`	Colocação do marcador "Marcador n°" em canal próprio, sem influência do atual processamento. SETM() permanece válido após o RESET e NC-START.

CLEARM (marcador n°, marcador n°,	Apaga o marcador "Marcador n°" em canal próprio, sem influência do atual processamento. Todos marcadores no canal podem ser apagados com CLEARM(). CLEARM (0) apaga o marcador "0". CLEARM() permanece válido após o RESET e NC-START.
n	Respectivo número de canal ou nome de canal

Indicação

Todos comandos mencionados acima precisam estar em blocos próprios.

O número de marcadores depende da CPU que estiver instalada.

Números de canal

Para os canais a serem coordenados podem ser especificados até 10 canais como número de canal (valor Integer).

Nomes de canal

Os nomes de canal devem ser convertidos em números através de variáveis (veja o capítulo "Variáveis e parâmetros de cálculo") ou, ao invés de números de canal, também podem ser programados os nomes de canal definidos através do $MC_CHAN_NAME (identificador ou palavra-chave). Os nomes definidos devem corresponder às convenções de linguagem NC (isto é, os primeiros dois caracteres devem compostos por letras ou um sublinhado).

⚠ CUIDADO

A atribuição dos números deve ser protegida contra alterações inadvertidas.

Os nomes não podem ter outro significado no NC como o de palavra-chave, comando de linguagem, nome de eixo, etc.

SETM() e CLEARM()

`SETM()` e `CLEARM()` também pode ser programados a partir de uma ação sincronizada. Veja o capítulo "Definir / deletar marcadores de espera: SETM CLEARM"

Exemplo

Canal com nome "MAQUINA" deve conter o número de canal 1,

Canal com nome "CARREGADOR" deve conter o número de canal 2:

```
DEF INT MAQUINA=1, CARREGADOR=2
```

As variáveis contém o mesmo nome que os canais.

Com isso a instrução `START` é:

```
START(MAQUINA)
```

Exemplo de coordenação de programa

Canal 1:

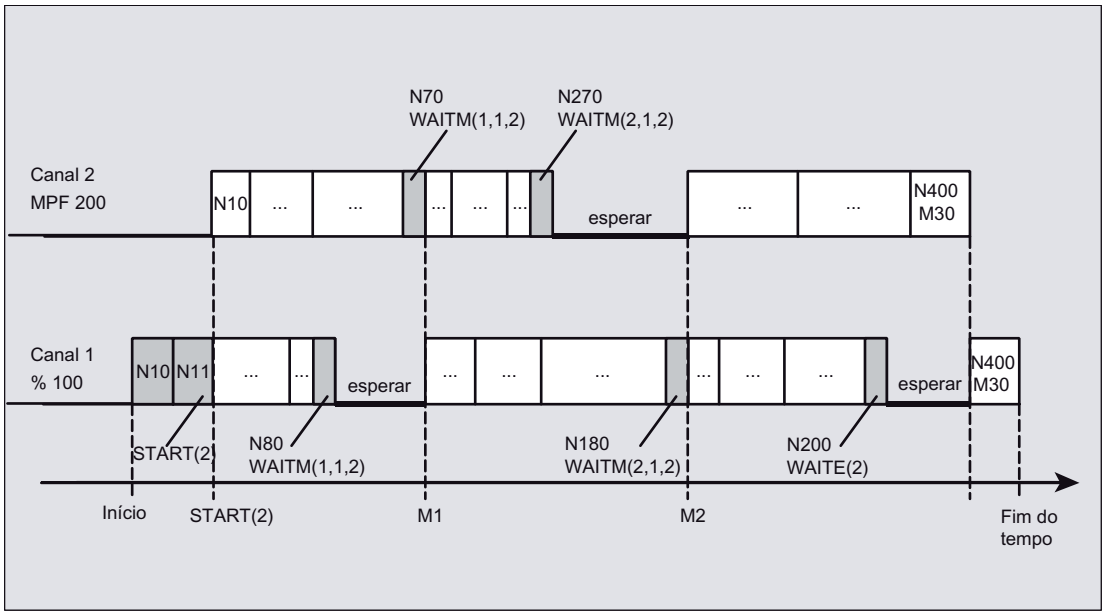
_N_MPF100_MPF

```
Código de programa          Comentário
N10 INIT(2,"MPF200")
N11 START(2)                ; Processamento no canal 2
...
N80 WAITM(1,1,2)            ; Espera pelo marcador WAIT 1 no canal 1 e
                              no canal 2 um outro processamento no canal 1
...
N180 WAITM(2,1,2)           ; Espera pelo marcador WAIT 2 no canal 1 e
                              no canal 2 um outro processamento no canal 1
...
N200 WAITE(2)               ; Espera pelo fim do programa do canal 2
N201 M30                    ; Fim do programa do canal 1, fim total
...
```

Canal 2:

_N_MPF200_MPF

```
Código de programa     Comentário
;$PATH=/_N_MPF_DIR
                       ; Processamento no canal 2
N70 WAITM(1,1,2)       ; Espera pelo marcador WAIT 1 no canal 1 e no canal 2 um
                         outro processamento no canal 1
...
N270 WAITM(2,1,2)      ; Espera pelo marcador WAIT 2 no canal 1 e no canal 2 um
                         outro processamento no canal 2
...
N400 M30               ; Fim do programa do canal 2
```



Exemplo: Programa a partir da peça

Código de programa
`N10 INIT(2,"/_N_WCS_DIR/_N_EIXO1_WPD/_N_DESBASTE1_MPF")`

Exemplo: Comando INIT com indicação relativa do caminho

No canal 1 está selecionado o programa `/_N_MPF_DIR/_N_MAIN_MPF`

Código de programa	**Comentário**
`N10 INIT(2,"MYPROG")`	`; Selecionar programa /_N_MPF_DIR/_N_MYPROG_MPF no canal 2`

Exemplo: Nome de canal e número de canal com variável Integer

```
$MC_CHAN_NAME[0]= "CHAN_X" ;Nome do 1º canal
$MC_CHAN_NAME[1]= "CHAN_Y" ;Nome do 2º canal
```

Código de programa	Comentário
START(1, 2)	; Executar a partida no 1° e 2° canal

Análogo à programação com os identificadores de canal:

Código de programa	Comentário
START(CHAN_X, CHAN_Y)	; Executar a partida no 1° e 2° canal ; Os identificadores Canal_X e Canal_Y representam internamente o número de canal 1 e 2 em função do dado de máquina $MC_CHAN_NAME. De acordo com isso eles também executam uma partida no 1° e 2° canal

Programação com variável Integer:

Código de programa	Comentário
DEF INT chanNo1, chanNo2)	; Definir número de canal
chanNo1=CHAN_X chanNo2=CHAN_Y	
START(chanNo1, chanNo2)	

1.14 Rotina de interrupção (ASUP)

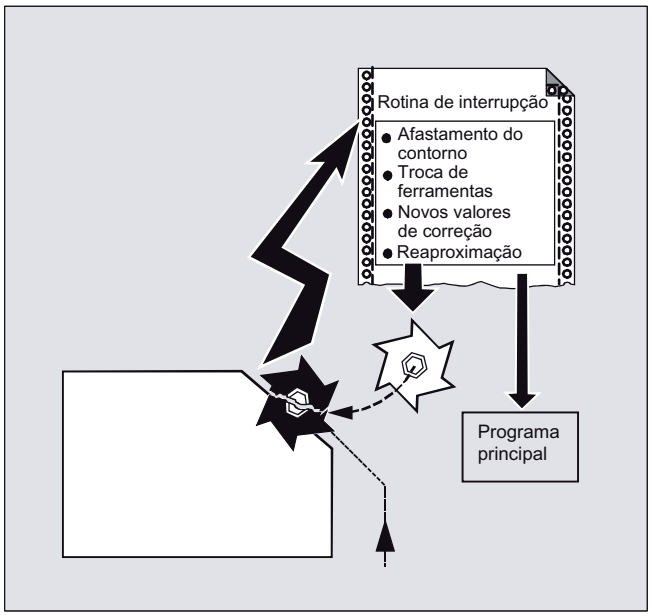
1.14.1 Funcionamento de uma rotina de interrupção

Indicação

Os termos "Subrotina assíncrona (ASUP)" e "Rotina de interrupção" que aparecem alternadamente na seguinte descrição significam a mesma funcionalidade.

Função

O funcionamento de uma rotina de interrupção deve ser explanada com base em um exemplo típico:



A ferramenta quebra durante a usinagem. Com isso se dispara um sinal que pára o processo de usinagem em andamento e ao mesmo tempo é iniciada uma subrotina – chamada de rotina de interrupção. Nesta subrotina estão contidas as instruções que deverão ser executadas para este caso.

Uma vez terminada a execução da subrotina (e com isso restabelecida a operabilidade), em função do comando `REPOS`, o comando retorna para o programa principal e continua com a usinagem a partir do ponto de interrupção (veja " Reaproximação do contorno (Página 488) ").

⚠ CUIDADO

Se na subrotina for programado nenhum comando `REPOS`, então se executa o posicionamento no ponto final do bloco que segue o bloco interrompido.

Literatura

Manual de funções básicas; BAG, Canal, Operação do programa, Comportamento Reset (K1), Capítulo: "Subrotinas assíncronas (ASUPs), Rotinas de interrupção"

1.14.2 Criar rotina de interrupção

Criar rotina de interrupção como subrotina

Durante a definição, a rotina de interrupção é identificada como se fosse uma subrotina.

Exemplo:

```
Código de programa          Comentário
PROC RETRAC_Z               ; Nome de programa "RETRAC_Z"
N10 ...                     ; Em seguida seguem os blocos NC.
...
N50 M17                     ; No final se coloca o fim do programa e o retorno ao
                              programa principal.
```

Salvar funções G modais (SAVE)

Na definição, a rotina de interrupção pode ser identificada com SAVE.

O atributo SAVE faz com que as funções G modais sejam salvas antes da chamada da rotina de interrupção e novamente ativadas no fim da subrotina de interrupção (veja " Subrotinas com mecanismo SAVE (SAVE) (Página 161) ").

Com isso é possível continuar a usinagem do ponto de interrupção depois da rotina de interrupção ser concluída.

Exemplo:

```
Código de programa
PROC RETRAC_Z SAVE
N10 ...
...
N50 M17
```

Atribuir outras rotinas de interrupção (SETINT)

Dentro da rotina de interrupção se pode programar instruções SETINT (veja "Atribuir e iniciar rotinas de interrupção (SETINT)" (Página 114)) e com isso podem ser ativadas outras rotinas de interrupção. A inicialização só ocorre com a entrada.

Literatura

Para mais informações sobre a criação de subrotinas, veja o capítulo "Técnica de subrotinas, técnica de macros".

1.14.3 Atribuição e partida de rotinas de interrupção (SETINT, PRIO, BLSYNC)

Função

O comando possui sinais (entrada 1…8) que desencadeiam uma interrupção do programa em andamento e que podem iniciar uma rotina de interrupção correspondente.

A atribuição de qual entrada pertence a qual programa é realizada no programa de peça através do comando SETINT.

Se no programa de peça existirem várias instruções SETINT e com isso mais sinais sendo disparados, então se deve atribuir valores de prioridade às rotinas de interrupção, que definem a seqüência durante a execução: PRIO=<valor>

Se ocorrerem novos sinais durante a execução da interrupção, as rotinas de maior prioridade cancelarão a atual rotina de interrupção.

Sintaxe

```
SETINT(<n>) PRIO=<valor> <NOME>
SETINT(<n>) PRIO=<valor> <NAME> BLSYNC
SETINT(<n>) PRIO=<valor> <NAME> LIFTFAST
```

Significado

SETINT(<n>):	Comando: Atribuir a entrada <n> de uma rotina de interrupção. A rotina de interrupção atribuída é iniciada assim que a entrada <n> for ativada. **Nota:** Se uma nova rotina for atribuída em uma entrada já ocupada, a atribuição anterior será desativada automaticamente.	
<n>:	Parâmetro: Número da entrada	
	Tipo:	INT
	Faixa de valores:	1 ... 8
PRIO= :	Comando: Definição da prioridade	
<valor>:	Valor de prioridade	
	Tipo:	INT
	Faixa de valores:	1 ... 128 A prioridade 1 corresponde ao mais alto nível de prioridade.
<NOME>:	Nome da subrotina (rotina de interrupção) que deve ser executada.	
BLSYNC:	Quando a instrução SETINT é programada junto com o BLSYNC, então com o disparo do sinal de interrupção ainda se executa o bloco de programa em andamento e, apenas depois disso que a rotina de interrupção é iniciada.	
LIFTFAST:	Quando a instrução SETINT é programada junto com o LIFTFAST, então com o disparo do sinal de interrupção é realizada uma "Retração rápida da ferramenta do contorno" ainda antes da partida da rotina de interrupção (veja " Retração rápida do contorno (SETINT LIFTFAST, ALF) (Página 118)").	

Exemplos

Exemplo 1: Atribuir rotinas de interrupção e definir prioridade

```
Código de programa                          Comentário
...
N20 SETINT(3) PRIO=1 RETRAC_Z               ; Se a entrada 3 for ativada, então deve ser
                                              iniciada a rotina de interrupção "RETRAC_Z".
N30 SETINT(2) PRIO=2 RETRAC_X               ; Se a entrada 2 for ativada, então deve ser
                                              iniciada a rotina de interrupção "RETRAC_X".
...
```

As rotinas de interrupção são executadas uma a uma na seqüência dos valores de prioridade, se as entradas forem disponibilizadas simultaneamente. primeiro "RETRAC_Z", depois "RETRAC_X".

Exemplo 2: Nova atribuição de rotinas de interrupção

```
Código de programa                          Comentário
...
N20 SETINT(3) PRIO=2 RETRAC_Z               ; Se a entrada 3 for ativada, então deve ser
                                              iniciada a rotina de interrupção "RETRAC_Z".
…
N120 SETINT(3) PRIO=1 RETRAC_X              ; Se atribui uma nova rotina de interrupção à
                                              entrada 3: ao invés de "RETRAC_Z" se deve
                                              iniciar "RETRAC_X", se for ativada a entrada
                                              3.
```

1.14.4 Desativar / ativar a atribuição de uma rotina de interrupção (DISABLE, ENABLE)

Função

Uma instrução SETINT é desativada com DISABLE e novamente ativada com ENABLE sem perder a associação Entrada → Rotina de interrupção.

Sintaxe

```
DISABLE(<n>)
ENABLE(<n>)
```

Significado

DISABLE(<n>):	Comando: **Desativação** da associação da rotina de interrupção com a entrada <n>
ENABLE(<n>):	Comando: **Reativação** da associação da rotina de interrupção com a entrada <n>
<n>:	Parâmetro: Número da entrada
	Tipo: INT
	Faixa de valores: 1 ... 8

Exemplo

Código de programa	Comentário
...	
N20 SETINT(3) PRIO=1 RETRAC_Z	; Se a entrada 3 for ativada, então deve ser iniciada a rotina de interrupção "RETRAC_Z".
...	
N90 DISABLE(3)	; A instrução SETINT do N20 é desativada.
...	
N130 ENABLE(3)	; A instrução SETINT do N20 é novamente ativada.
...	

1.14.5 Deletar a atribuição de uma rotina de interrupção (CLRINT)

Função

Uma associação Entrada → Rotina de interrupção definida com `SETINT` pode ser desfeita com `CLRINT`.

Sintaxe

```
CLRINT(<n>)
```

Significado

`CLRINT(<n>)`:	Comando: Deletação da associação da rotina de interrupção com a entrada **<n>**
`<n>`:	Parâmetro: Número da entrada
	Tipo: INT
	Faixa de valores: 1 ... 8

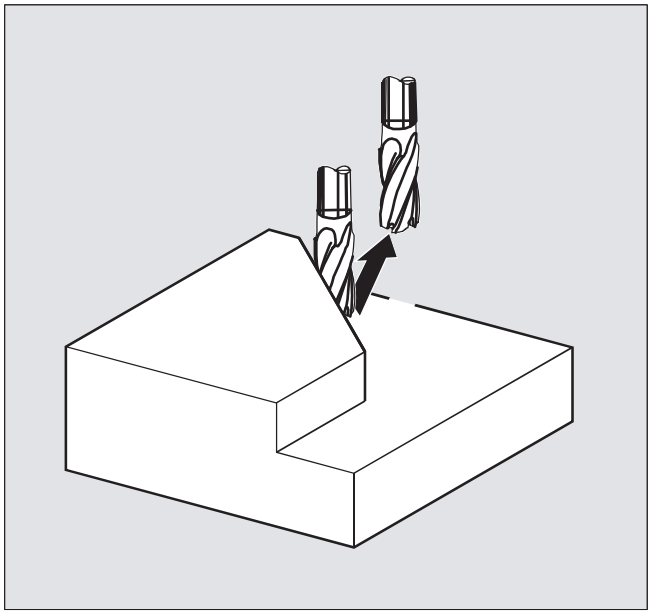
Exemplo

Código de programa	**Comentário**
...	
N20 SETINT(3) PRIO=2 RETRAC_Z	;
...	
N50 CLRINT(3)	; A associação entre a entrada "3" e a rotina de interrupção "RETRAC_Z" foi deletada.
...	

1.14.6 Retração rápida do contorno (SETINT LIFTFAST, ALF)

Função

Através de uma instrução `SETINT` com `LIFTFAST` se realiza o afastamento da ferramenta mediante retração rápida assim que a entrada for ativada.



A execução restante depende se a instrução `SETINT`, além da `LIFTFAST`, também contiver uma rotina de interrupção:

Com rotina de interrupção:	A rotina de interrupção é executada **após** a retração rápida.
Sem rotina de interrupção:	A usinagem é parada com alarme após a retração rápida.

Sintaxe

```
SETINT(<n>) PRIO=1 LIFTFAST
SETINT(<n>) PRIO=1 <NAME> LIFTFAST
```

Significado

`SETINT(<n>)`:	Comando: Atribuir a entrada <n> de uma rotina de interrupção. A rotina de interrupção atribuída é iniciada assim que a entrada <n> for ativada.
<n>:	Parâmetro: Número da entrada
	Tipo: INT
	Faixa de valores: 1 ... 8
`PRIO=` :	Definição da prioridade

<valor>:	Valor de prioridade
	Faixa de valores: 1 ... 128
	A prioridade 1 corresponde ao mais alto nível de prioridade.
<NOME>:	Nome da subrotina (rotina de interrupção) que deve ser executada.
LIFTFAST:	Comando: Retração rápida do contorno
ALF=… :	Comando: Sentido de deslocamento programável (no bloco de deslocamento)
	Para conhecer as opções de programação com ALF, veja o assunto " Sentido de deslocamento na retração rápida do contorno (Página 120) ".

Condições gerais

Comportamento com Frame ativo com espelhamento

Na determinação do sentido de retração é verificado se um Frame está ativo com espelhamento. Neste caso a direita e a esquerda são invertidas para o sentido de retração em função do sentido de tangente. Os componentes de sentido no sentido da ferramenta não são espelhados. Este comportamento é ativado através do ajuste de dado de máquina:

MD21202 $MC_LIFTFAST_WITH_MIRROR = TRUE

Exemplo

Uma ferramenta cancelada deve ser substituída automaticamente por uma ferramenta gêmea. A usinagem é continuada com a nova ferramenta.

Programa principal:

Programa principal	**Comentário**
N10 SETINT(1) PRIO=1 TROCA_F LIFTFAST	; Quando a entrada 1 é ativada, a ferramenta é imediatamente afastada do contorno com retração rápida (código n° 7 para correção do raio da ferramenta G41). Em seguida é executada a rotina de interrupção "TROCA_F".
N20 G0 Z100 G17 T1 ALF=7 D1	
N30 G0 X-5 Y-22 Z2 M3 S300	
N40 Z-7	
N50 G41 G1 X16 Y16 F200	
N60 Y35	
N70 X53 Y65	
N90 X71.5 Y16	
N100 X16	
N110 G40 G0 Z100 M30	

Subrotina:

Subrotina	Comentário
PROC TROCA_F SAVE	; Subrotina com armazenamento do atual estado operacional
N10 G0 Z100 M5	; Posição de troca de ferramentas, parada do fuso
N20 T11 M6 D1 G41	; Trocar ferramentas
N30 REPOSL RMB M3	; Reaproximação do contorno e salto de retorno para o programa principal (se programa em um bloco próprio)

1.14.7 Sentido de deslocamento na retração rápida do contorno

Movimento de retrocesso

O plano do movimento de retrocesso é definida através do seguinte código G:

- LFTXT

  O plano do movimento de retrocesso é determinado a partir da tangente da trajetória e do sentido da ferramenta (ajuste padrão).

- LFWP

  O plano do movimento de retrocesso é o plano de trabalho ativo que se seleciona com o código G17, G18 ou G19. O sentido do movimento de retrocesso não depende da tangente da trajetória. Com isso pode-se programar uma retração rápida paralela ao eixo.

- LFPOS

  Retrocesso do eixo identificado com POLFMASK / POLFMLIN na posição de eixo absoluta programada com POLF.

  O ALF não tem nenhuma influência sobre o sentido de retração para vários eixos assim como para vários eixos de contexto linear.

  Literatura:
  Manual de programação Fundamentos; capítulo: "Retrocesso rápido para rosqueamento"

Sentido de deslocamento programável (ALF=...)

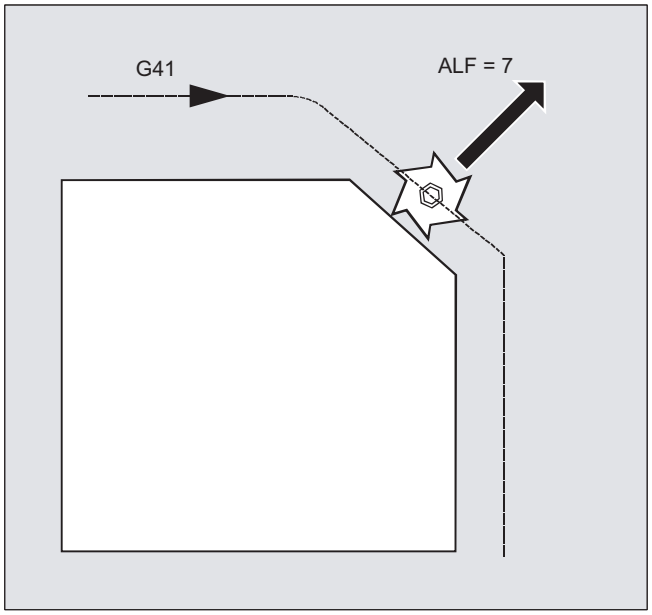
No plano do movimento de retrocesso o sentido é programado com ALF, em discretos passos de 45 graus.

Os possíveis sentidos de deslocamento estão armazenados no comando numérico através de códigos especiais e eles podem ser chamados através deste número.

Exemplo:

Código de programa
N10 SETINT(2) PRIO=1 RETRAC_Z LIFTFAST
ALF=7

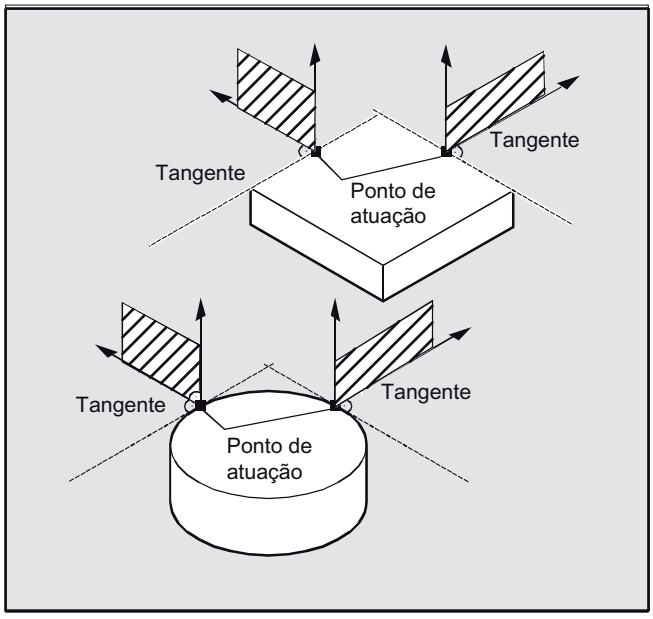
Com o `G41` (sentido de usinagem à esquerda do contorno) ativado, a ferramenta se afasta perpendicularmente do contorno.



Plano de referência para a descrição dos sentidos de deslocamento com LFTXT

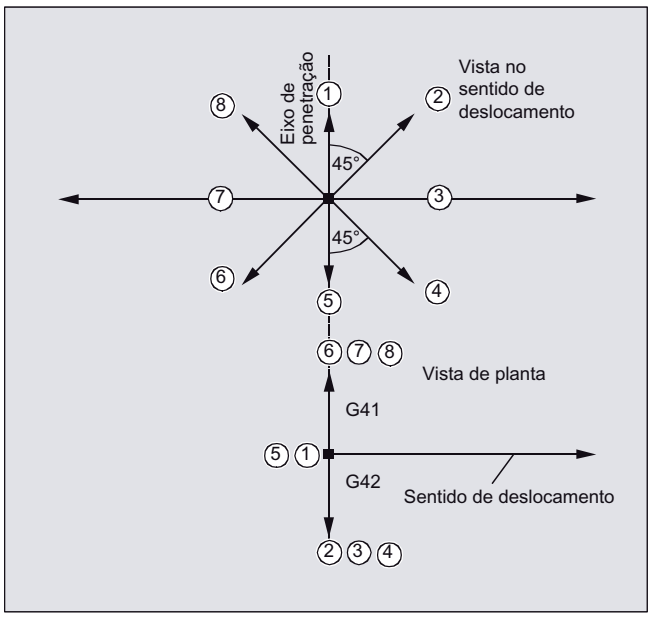
No ponto de aplicação da ferramenta no contorno programado é fixado um plano que serve de referência para a especificação do movimento de retração com o respectivo código.

O plano de referência é fixado a partir de um eixo longitudinal da ferramenta (sentido de penetração) e um vetor que está posicionado perpendicularmente a esse eixo e para o ponto de aplicação da ferramenta no contorno.



Códigos com sentidos de deslocamento com LFTXT

Os números de código com os sentidos de deslocamento a partir do plano de referência estão indicados na figura a seguir.



Para `ALF=1` se define o retrocesso no sentido da ferramenta.

Com `ALF=0` se desativa a função de "Retração rápida".

⚠ CUIDADO

Com a correção do raio da ferramenta ativada:

- os códigos 2, 3, 4 com `G41`
- os códigos 6, 7, 8 com `G42`

não devem ser utilizados, pois nestes casos a ferramenta poderia colidir com a peça enquanto se dirige até o contorno.

Códigos com sentidos de deslocamento com LFWP

Com LFWP o sentido do plano de trabalho é atribuído como segue:

- G17: Plano X/Y

  ALF=1: Retrocesso no sentido X

  ALF=3: Retrocesso no sentido Y

- G18: Plano Z/X

  ALF=1: Retrocesso no sentido Z

  ALF=3: Retrocesso no sentido X

- G19: Plano Y/Z

  ALF=1: Retrocesso no sentido Y

  ALF=3: Retrocesso no sentido Z

1.14.8 Execução de movimentos com rotinas de interrupção

Rotina de interrupção sem LIFTFAST

Os movimentos dos eixos são desacelerados na trajetória até pararem. Em seguida é iniciada a rotina de interrupção.

A posição de parada é armazenada (memorizada) como posição de interrupção e será novamente aproximada com REPOS com RMI no final da rotina de interrupção.

Rotina de interrupção com LIFTFAST

Os movimentos dos eixos são freados na trajetória. Ao mesmo tempo se executa o movimento LIFTFAST como movimento sobreposto. A rotina de interrupção é iniciada quando for parado o movimento de percurso e o movimento LIFTFAST.

Como posição de interrupção é memorizada a posição no contorno onde foi iniciado o movimento LIFTFAST, onde o percurso também foi abandonado.

Com LIFTFAST e ALF=0 a rotina de interrupção se comporta de modo idêntico como a rotina de interrupção sem LIFTFAST.

Indicação

O valor com que os eixos geométricos se afastam do contorno na retração rápida, pode ser ajustado através de um dado de máquina.

1.15 Troca de eixos, troca de fusos (RELEASE, GET, GETD)

Função

Um ou vários eixos ou fusos sempre podem interpolados em apenas um canal. Se um eixo precisa alternar entre dois canais diferentes (p. ex. trocador de paletes), então se deve liberar primeiro o atual canal e depois passado para o outro canal. O eixo é trocado entre os canais.

Ampliação da troca de eixos

Um eixo/fuso pode ser trocado com parada de pré-processamento e aça sincronizada entre o pré-processamento e o processamento principal, ou também, alternativamente sem a parada de pré-processamento. Além disso, uma troca de eixos também é possível através de

- Rotação de contentor de eixo AXCTSWE ou AXCTWED mediante `GET/GETD` implícito.
- Frame com rotação, se através disso este eixo estiver ligado com outro eixo.
- Ações sincronizadas; veja em ações sincronizadas de movimentos, "Troca de eixos `RELEASE`, `GET`".

Fabricante da máquina

Observe as instruções do fabricante da máquina. Através de dados de máquina configuráveis, um eixo deve ser definido de forma clara em todos os canais para a troca de eixos, e o comportamento de troca de eixos também podem ser ajustado de forma modificável através de dados de máquina.

Sintaxe

`RELEASE(nome de eixo, nome de eixo, ...)` ou `RELEASE(S1)`

`GET(nome de eixo, nome de eixo, ...)` ou `GET(S2)`

`GETD(nome de eixo, nome de eixo, ...)` ou `GETD(S3)`

Com GETD (GET Directly) um eixo é buscado diretamente de outro canal. Isto significa que para este GETD não é necessário programar um RELEASE adequado em outro canal. Isto também significa que agora se deve estabelecer outra comunicação de canais (p. ex. marcadores Wait).

Significado

`RELEASE (nome de eixo, nome de eixo, …)`:	Habilitação do eixo (n)
`GET (nome de eixo, nome de eixo, …)`:	Aceitação do eixo(n)
`GETD (nome de eixo, nome de eixo, …)`:	Aceitação direta do eixo(n)
`Nome de eixo`:	Atribuição de eixos no sistema: AX1, AX2, … ou especificação dos nomes de eixo da máquina
`RELEASE(S1)` :	Liberação dos fusos S1, S2, …
`GET(S2)` :	Aceitação dos fusos S1, S2, …
`GETD(S3)` :	Aceitação direta dos fusos S1, S2, …

Solicitação GET sem parada de pré-processamento

Se, após uma solicitação GET **sem** parada de pré-processamento, o eixo for novamente habilitado com `RELEASE(eixo)` ou `WAITP(eixo)`, então um `GET` seguinte conduzirá para um `GET` **com** parada de pré-processamento.

⚠ CUIDADO

Um eixo aceito com GET permanece atribuído neste canal, mesmo após uma tecla ou programa RESET.

Com uma inicialização do programa a atribuição dos eixos ou fusos trocados deverá ocorrer via programa, caso o eixo não seja necessário em seu canal básico.

Com POWER ON ele será atribuído ao canal definido nos dados da máquina.

Exemplos

Exemplo 1: Troca entre dois canais

Para o processamento no canal 1 são utilizados de 6 eixos os seguintes eixos: 1., 2., 3. e 4º eixo.
Os 5º e 6º eixos são utilizados no canal 2 para a troca de peças de trabalho.

O eixo 2 deve ser trocado entre dois canais e depois do POWER ON ser associado ao canal 1.

Programa "MAIN" no canal 1:

```
Código de programa              Comentário
INIT (2,"TROCA2")               ; Selecionar o programa TROCA2 no canal 2.
N… START (2)                    ; Iniciar o programa no canal 2.
N… GET (AX2)                    ; Aceitar eixo AX2.
...
N… RELEASE (AX2)                ; Liberar eixo AX2.
N… WAITM (1,1,2)                ; Espera pelo marcador WAIT no canal 1 e 2 para
                                  sincronização em ambos canais.
...                             ; Continuação do processo após a troca de eixos.
N… M30
```

Programa "TROCA2" no canal 2:

```
Programação                     Comentário
N… RELEASE(AX2)
N160 WAITM(1,1,2)               ; Espera pelo marcador WAIT no canal 1 e 2 para
                                  sincronização em ambos canais.
N150 GET(AX2)                   ; Aceitar eixo AX2.
...                             ; Continuação do processo após a troca de eixos.
N… M30
```

Exemplo 2: Troca de eixos sem sincronização

Se o eixo não precisa ser sincronizado, não é criada nenhuma parada de pré-processamento pelo GET.

```
Programação                    Comentário
N01 G0 X0
N02 RELEASE(AX5)
N03 G64 X10
N04 X20
N05 GET(AX5)                   ; Se nenhuma sincronização for necessária, este não é
                                 um bloco executável.
N06 G01 F5000                  ; Nenhum bloco executável.
N07 X20                        ; Nenhum bloco executável, pois a posição X é igual
                                 no N04.
N08 X30                        ; Primeiro bloco executável após N05.
...
```

Exemplo 3: Ativação de uma troca de eixos sem parada de pré-processamento

Requisito: A troca de eixos sem parada de pré-processamento deve ser configurada através de um dado de máquina.

```
Programação                    Comentário
N010 M4 S100
N011 G4 F2
N020 M5
N021 SPOS=0
N022 POS[B]=1
N023 WAITP(B)                  ; O eixo B passa a ser eixo neutro.
N030 X1 F10
N031 X100 F500
N032 X200
N040 M3 S500                   ; O eixo não dispara nenhuma parada de pré-
                                 processamento/REORG.
N041 G4 F2
N050 M5
N099 M30
```

Se o fuso ou eixo B for deslocado imediatamente após o bloco N023 como **eixo PLC** p. ex. desloca 180 graus e depois retorna até 1 grau, então este eixo novamente passa a ser um eixo neutro e não dispara nenhuma parada de pré-processamento no bloco `N40`.

Requisitos

Condições para a troca de eixos

- O eixo deverá estar definido em dados da máquina para todos canais por ele utilizados.
- Através do dado de máquina específico de **eixo** deverá estar definido qual canal o eixo deve ser atribuído após POWER ON.

Descrição

Liberar eixo: RELEASE

Na liberação do eixo deve-se observar:

1. O eixo não pode participar de nenhuma transformação.
2. Nos acoplamentos de eixo (comando tangencial) todos eixos envolvidos precisam ser liberados.
3. Um eixo de posicionamento concorrente não pode ser trocado neste estado.
4. Em um eixo mestre Gantry também ocorre a troca para todos eixos escravos.
5. Em acoplamentos de eixo (movimento acoplado, acoplamento de valor mestre, transmissão eletrônica) somente pode ser liberado o eixo mestre do grupo.

Aceitar eixo: GET

Com este comando é executada a troca de eixos propriamente dita. A responsabilidade para o eixo está totalmente no canal onde foi programado o comando.

Efeitos do GET:

Troca de eixos com sincronização:

Um eixo sempre precisa ser sincronizado quando ele estava atribuído em um outro canal ou no PLC, e se antes do GET ocorreu uma sincronização pelo "WAITP", G74 ou se foi anulado o curso restante.

- Ocorre uma parada de pré-processamento (como no SOPRE)
- O processamento permanece interrompido até que a troca seja completamente executada.

"GET" automático

Se a princípio um eixo está disponível no canal, mas nesse momento não como "Eixo de canal", o GET é executado automaticamente. Se o eixo ou os eixos já estão sincronizados, não é criada nenhuma parada de pré-processamento.

Ajustar o comportamento de eixos de forma modificável

O momento de entrega dos eixos pode ser ajustado através de um dado de máquina da seguinte forma:

- A troca automática de eixos entre dois canais também é executada se o eixo tiver sido colocado em estado neutro (transferência convencional) com WAITP
- Na solicitação de uma rotação de contentor de eixos todos os eixos do contentor que estiverem atribuídos ao canal executado serão chamados no canal através do GET ou GETD implícito. Uma troca de eixos seguinte somente será permitida após a conclusão da rotação do contentor de eixos.
- Após um bloco intermediário colocado no processamento principal se controla se uma reorganização é ou não necessária. Uma reorganização somente se faz necessária quando os estados dos eixos deste bloco **não** coincidirem com os atuais estados dos eixos.
- Ao invés de um bloco GET com parada de pré-processamento e ação sincronizada entre o pré-processamento e o processamento principal, também pode ser realizada uma troca de eixos sem o pré-processamento. Então apenas se gera um bloco intermediário com a solicitação GET. No processamento principal, durante a execução deste bloco, se controla se os estados do eixo no bloco estão de acordo com os estados atuais de eixo.

Para mais informações sobre a funcionalidade de uma troca de eixos ou de fusos, veja o /FB2/ Manual de funções ampliadas; BAGs, Canais, Troca de eixos (K5).

1.16 Transferir eixo de outro canal (AXTOCHAN)

Função

Com o comando de linguagem AXTOCHAN se pode solicitar um eixo que será transferido para outro canal. O eixo pode ser transferido ao canal correspondente tanto do programa de peça NC como de uma ação sincronizada.

Sintaxe

```
AXTOCHAN(nome de eixo,número de canal[,nome de eixo,número de
canal[,...]])
```

Significado

AXTOCHAN:	Solicitar eixo para um determinado canal
Nome de eixo:	Atribuição de eixos no sistema: X, Y, … ou especificação do nome de eixo de máquina envolvido. O canal a ser executado não precisa ser o próprio canal e ele também não precisa ser o canal que atualmente detém o direito de interpolação do eixo
Número de canal:	Número do canal que deve ser atribuído ao eixo

Indicação

Eixo de posicionamento concorrente e eixo controlado exclusivamente por PLC

Um eixo PLC não pode mudar de canal como eixo de posicionamento concorrente. Um eixo controlado exclusivamente pelo PLC não pode ser atribuído ao programa NC.

Literatura
Manual de funções ampliadas; Eixos de posicionamento (P2)

Exemplo

AXTOCHAN no programa NC

Os eixos X e Y são conhecidos no 1º canal e no 2º canal. Atualmente o canal 1 detém o direito de interpolação e no canal 1 se inicia o seguinte programa:

```
Código de programa              Comentário
N110 AXTOCHAN(Y,2)              ; Transferir o eixo Y para o 2° canal.
N111 M0
N120 AXTOCHAN(Y,1)              ; Busca novamente o eixo B (neutro).
N121 M0
N130 AXTOCHAN(Y,2,X,2)          ; Transferir o eixo Y e o eixo X para o 2° canal
                                  (eixos neutros).
N131 M0
N140 AXTOCHAN(Y,2)              ; Transferir o eixo Y para o 2° canal (programa NC).
N141 M0
```

Outras informações

AXTOCHAN no programa NC

Neste caso, apenas com uma solicitação do eixo para o programa NC em canal próprio é que se executa um GET e com isso também se espera pela real alteração de estados. Se o eixo for solicitado para outro canal ou se ele deve ser um eixo neutro no próprio canal, então apenas será cancelada a solicitação.

AXTOCHAN a partir de uma ação sincronizada

Se um eixo for solicitado para o próprio canal, então o AXTOCHAN de uma ação sincronizada forma o GET a partir de uma ação sincronizada. Neste caso o eixo passa a ser o eixo neutro na primeira solicitação do próprio canal. Na segunda solicitação o eixo do programa NC é atribuído de forma análoga à solicitação GET no programa NC. Para solicitação GET a partir de uma ação sincronizada, veja o capítulo "Ações sincronizadas de movimentos".

1.17 Ativar dados de máquina (NEWCONF)

Função

Com o comando NEWCONF são ativados todos os dados de máquina de nível de atividade "NEW_CONFIG". A função também pode ser ativada na interface de operação HMI através da softkey "Ativar MD".

Com a execução da função "NEWCONF" é executada uma parada de pré-processamento implícita, isto é, se interrompe o movimento de percurso.

Sintaxe

```
NEWCONF
```

Significado

NEWCONF: Comando para tornar ativos todos os dados de máquina do nível de ativação "NEW_CONFIG"

Executar o NEWCONF a partir de um programa de peça entre vários canais

Se dados de máquina de eixos forem alterados a partir do programa de peça e, sem seguida, ativados com NEWCONF, então o NEWCONF ativa apenas os dados de máquina que realizam alterações para o canal do programa de peça.

Indicação

Para permitir que todas alterações sejam ativadas, se deve executar o comando NEWCONF em todo canal onde atualmente são calculados os eixos ou funções alterados que foram afetados pelos dados de máquina.

Com NEWCONF não se ativa nenhum dado de máquina de eixos.

Para eixos controlados por PLC deve ser executado um RESET de eixo.

Exemplo

Fresamento: Execução da posição de furação com diferentes tecnologias

Código de programa	Comentário
N10 $MA_CONTOUR_TOL[AX]=1.0	; Modificar dados de máquina.
N20 NEWCONF	; Ativar dados de máquina.
...	

1.18 Gravar arquivo (WRITE)

Função

Com o comando WRITE os blocos/dados podem ser gravados no fim em um arquivo indicado (arquivo de protocolo) a partir do programa de peça que está em execução. Os blocos/dados são inseridos no final do arquivo, após o M30.

Indicação

Um arquivo que deve ser gravado com o comando WRITE será criado, caso ele ainda não exista no NC.

O local de armazenamento é a memória NC estática. Para o SINUMERIK 840D sl é o cartão CompactFlash Card. Em comparação ao SINUMERIK 840D o tempo de processamento do comando WRITE aumenta em aprox. 75 ms.

Se já existir um arquivo de mesmo nome no disco rígido, ele será substituído pelo novo assim que o arquivo for fechado (no NC). Solução: Modifique o nome no NC com a sotkey "Propriedades" na área de operação "Serviços".

Pré-requisito

O atual nível de proteção ajustado deve ser igual ou mais alto do que o direito READ do arquivo. Se este não for o caso, o acesso será negado com mensagem de erro (valor retornado das variáveis com erro = 13).

Sintaxe

```
DEF INT <erro>
WRITE(<erro>,"<nome de arquivo>","<bloco/dados>")
```

Significado

WRITE:	Comando para adicionar um bloco ou dados no fim do arquivo indicado		
<erro>:	Variável para o retorno do valor do erro		
	Tipo.	INT	
	Valor:	0	nenhum erro
		1	Caminho não permitido
		2	Caminho não encontrado
		3	Arquivo não encontrado
		4	Tipo de arquivo incorreto
		10	Arquivo está cheio
		11	Arquivo está sendo usado
		12	Nenhum ressource livre
		13	Sem direito de acesso
		20	Outros erros

<nome do arquivo>: Nome do arquivo onde o bloco especificado ou os dados especificados devem ser adicionados

Tipo: STRING

Ao indicar o nome do arquivo devem ser observados os seguintes itens:

- O nome de arquivo indicado não pode conter espaços vazios ou caracteres de controle (caracteres de código ASCII ≤ 32), caso contrário o comando WRITE será cancelado com o identificador de erro 1 "Caminho não permitido".
- O nome do arquivo pode ser especificado com indicação do caminho e extensão de arquivo:
  - Indicações de caminho

    As indicações de caminho devem ser absolutas, isto é, elas começam com "/".

    Sem a indicação do caminho o arquivo será armazenado no atual diretório (=diretório do programa selecionado).
  - Extensão de arquivo

    Se o nome de arquivo não possui nenhum identificador de domínio ("_N_"), ele será complementado de acordo.

    Se no nome de arquivo o quarto último caractere for um sublinhado "_", então os três caracteres seguintes serão interpretados como extensão de arquivo. Para utilizar o mesmo nome de arquivo em todos comandos de arquivo, por exemplo, através de uma variável do tipo STRING, somente podem ser utilizadas as extensões de arquivo _SPF e _MPF.

    Se nenhuma extensão "_MPF" ou "_SPF" for especificada, é complementado automaticamente com _MPF.
- O nome do arquivo não pode ter mais de 32 bytes, o tamanho da indicação do caminho não pode ter mais de 128 bytes.

Exemplo:

```
"PROTFILE"
"_N_PROTFILE"
"_N_PROTFILE_MPF"
"/_N_MPF_DIR_/_N_PROTFILE_MPF/"
```

<bloco/dados>: Bloco ou dados que devem ser adicionados no arquivo indicado.

Tipo: STRING

Nota:
Internamente ainda se adiciona o LF, ou seja, a seqüência de caracteres fica 1 caractere mais longa.

Condições gerais

- **Tamanho de arquivo máximo (→ Fabricante da máquina!)**

  O tamanho de arquivo máximo possível para arquivos de protocolo é ajustado com o dado de máquina:

  MD11420 $MN_LEN_PROTOCOL_FILE

  O tamanho de arquivo máximo vale para todos arquivos que foram criados com o comando `WRITE`. No caso de ser excedido, aparece uma mensagem de erro e o bloco ou os dados não serão armazenados. Enquanto houver espaço de memória, ainda se pode criar um novo arquivo.

Exemplos

Exemplo 1: Comando WRITE sem indicação de caminho absoluta

Código de programa	Comentário
N10 DEF INT ERROR	; Definição das variáveis de erro.
N20 WRITE(ERROR,"TEST1","PROTOCOLO DE 7.2.97")	; Grava o texto do "PROTOCOLO DE 7.2.97" no arquivo _N_TEST1_MPF.
N30 IF ERROR	; Avaliação de erro.
N40 MSG ("Erro no comando WRITE:" <<ERROR)	
N50 M0	
N60 ENDIF	
...	

Exemplo 2: Comando WRITE com indicação de caminho absoluta

Código de programa
...
WRITE(ERROR,"/_N_WCS_DIR/_N_PROT_WPD/_N_PROT_MPF","PROTOCOLO DE 7.2.97")
...

1.19 Deletar arquivo (DELETE)

Função

Com o comando DELETE se pode deletar todos os arquivos, independente deles serem criados ou não com o comando WRITE. Mesmo os arquivos atribuídos com um nível de acesso mais alto podem ser deletados com DELETE.

Sintaxe

```
DEF INT <erro>
DELETE(<erro>,"<nome de arquivo>")
```

Significado

DELETE:	Comando para deletar o arquivo indicado		
<erro>:	Variável para o retorno do valor do erro		
	Tipo.	INT	
	Valor:	0	nenhum erro
		1	Caminho não permitido
		2	Caminho não encontrado
		3	Arquivo não encontrado
		4	Tipo de arquivo incorreto
		11	Arquivo está sendo usado
		12	Nenhum resource livre
		20	Outros erros

`<nome do arquivo>`:

Nome do arquivo que deve ser deletado

Tipo: STRING

Ao indicar o nome do arquivo devem ser observados os seguintes itens:

- O nome de arquivo indicado não pode conter espaços vazios ou caracteres de controle (caracteres de código ASCII ≤ 32), caso contrário o comando `DELETE` será cancelado com o identificador de erro 1 "Caminho não permitido".
- O nome do arquivo pode ser especificado com indicação do caminho e extensão de arquivo:
  - Indicações de caminho

    As indicações de caminho devem ser absolutas, isto é, elas começam com "/".

    Sem a indicação do caminho o arquivo será procurado no atual diretório (=diretório do programa selecionado).
  - Extensão de arquivo

    Se o nome de arquivo não possui nenhum identificador de domínio ("_N_"), ele será complementado de acordo.

    Se no nome de arquivo o quarto último caractere for um sublinhado "_", então os três caracteres seguintes serão interpretados como extensão de arquivo. Para utilizar o mesmo nome de arquivo em todos comandos de arquivo, por exemplo, através de uma variável do tipo STRING, somente podem ser utilizadas as extensões de arquivo _SPF e _MPF.

    Se nenhuma extensão "_MPF" ou "_SPF" for especificada, é complementado automaticamente com _MPF.
- O nome do arquivo não pode ter mais de 32 bytes, o tamanho da indicação do caminho não pode ter mais de 128 bytes.

Exemplo:

```
"PROTFILE"
"_N_PROTFILE"
"_N_PROTFILE_MPF"
"/_N_MPF_DIR_/_N_PROTFILE_MPF/"
```

Exemplo

Código de programa	Comentário
N10 DEF INT ERROR	; Definição das variáveis de erro.
N15 STOPRE	; Parada de pré-processamento.
N20 DELETE(ERROR,"/_N_SPF_DIR/_N_TEST1_SPF")	; Deleta o arquivo TEST1 no diretório de subrotinas.
N30 IF ERROR	; Avaliação de erro.
N40 MSG ("Erro no comando DELETE:" <<ERROR)	
N50 M0	
N60 ENDIF	

1.20 Ler linhas no arquivo (READ)

Função

O comando `READ` lê uma ou mais linhas no arquivo especificado e armazena as informações lidas em um campo de tipo STRING. Cada linha lida ocupa um elemento deste campo.

Indicação

O arquivo precisa estar na memória de usuário estática do NCK (sistema de arquivos passivo).

Pré-requisito

O atual nível de proteção ajustado deve ser igual ou mais alto do que o direito READ do arquivo. Se este não for o caso, o acesso será negado com mensagem de erro (valor retornado das variáveis com erro = 13).

Sintaxe

```
DEF INT <erro>
DEF STRING[<tamanho de String>] <resultado>[<n>,<m>]
READ(<erro>,"<nome de arquivo>",<linha inicial>,<número de
linhas>,<resultado>)
```

Significado

`READ`: Comando para leitura de linhas do arquivo especificado e para armazenamento destas linhas em um campo de variável.

`<erro>`: Variável para o retorno do valor de erro (parâmetro Call-By-Reference)

Tipo. INT

Valor:

0	nenhum erro
1	Caminho não permitido
2	Caminho não encontrado
3	Arquivo não encontrado
4	Tipo de arquivo incorreto
13	Direito de acesso insuficiente
21	Linha não existente (parâmetro `<linha inicial>` ou `<número de linhas>` maior que o número de linhas no arquivo indicado).
22	O tamanho de campo das variáveis de resultado (`<resultado>`) é muito pequeno.
23	Área de linhas muito grande (o parâmetro `<número de linhas>` deve ser configurado de modo que seja permitida a leitura em toda extensão até o fim do arquivo).

<nome do arquivo>: Nome do arquivo que deve ser lido (parâmetro Call-By-Value)

Tipo: STRING

Ao indicar o nome do arquivo devem ser observados os seguintes itens:

- O nome de arquivo indicado não pode conter espaços vazios ou caracteres de controle (caracteres de código ASCII ≤ 32), caso contrário o comando READ será cancelado com o identificador de erro 1 "Caminho não permitido".
- O nome do arquivo pode ser especificado com indicação do caminho e extensão de arquivo:
  - Indicações de caminho

    As indicações de caminho devem ser absolutas, isto é, elas começam com "/".

    Sem a indicação do caminho o arquivo será procurado no atual diretório (=diretório do programa selecionado).
  - Extensão de arquivo

    Se o nome de arquivo não possui nenhum identificador de domínio ("_N_"), ele será complementado de acordo.

    Se no nome de arquivo o quarto último caractere for um sublinhado "_", então os três caracteres seguintes serão interpretados como extensão de arquivo. Para utilizar o mesmo nome de arquivo em todos comandos de arquivo, por exemplo, através de uma variável do tipo STRING, somente podem ser utilizadas as extensões de arquivo _SPF e _MPF.

    Se nenhuma extensão "_MPF" ou "_SPF" for especificada, é complementado automaticamente com _MPF.
- O nome do arquivo não pode ter mais de 32 bytes, o tamanho da indicação do caminho não pode ter mais de 128 bytes.

Exemplo:

```
"PROTFILE"
"_N_PROTFILE"
"_N_PROTFILE_MPF"
"/_N_MPF_DIR_/_N_PROTFILE_MPF/"
```

<linha inicial>: Linha inicial da área do arquivo que deve ser lida (parâmetro Call-By-Value)

Tipo: INT

Valor:		
	0	Antes do fim do arquivo é realizada a leitura do número de linhas especificado com o parâmetro <número de linhas>.
	1 … n	Número da primeira linha a ser lida.

<número de linhas>: Número de linhas que devem ser lidas (parâmetro Call-By-Value)

Tipo: INT

<resultado>: Variável de resultado (parâmetro Call-By-Reference)

Campo de variável onde o texto lido é armazenado.

Tipo: STRING (tamanho máx.: 255)

Se no parâmetro <número de linhas> forem especificadas menos linhas que o tamanho do campo [<n>,<m>] das variáveis de resultado, então os elementos de campo restantes não serão alterados.

A terminação de uma linha com o caractere de controle "LF" (Line Feed) ou "CR LF" (Carriage Return Libe Feed) **não** é armazenada na variável de resultado.

As linhas lidas serão truncadas se a linha for maior que o tamanho de String definido. Não se emite nenhuma mensagem de erro.

Indicação

Arquivos binários não podem ser lidos. É retornado o erro "tipo de arquivo incorreto" (valor de retorno da variável de erro = 4). Não se pode ler os seguintes tipos de arquivo: _BIN, _EXE, _OBJ, _LIB, _BOT, _TRC, _ACC, _CYC, _NCK.

Exemplo

Código de programa	Comentário
N10 DEF INT ERROR	; Definição das variáveis de erro.
N20 DEF STRING[255] RESULT[5]	; Definição das variáveis de resultado.
N30 READ(ERROR,"/_N_CST_DIR/_N_TESTFILE_MPF",1,5,RESULT)	; Nome de arquivo com identificadores de domínio e de arquivo e indicação do caminho.
N40 IF ERROR <>0	; Avaliação de erro.
N50 MSG("ERRO"<<ERROR<<"COM COMANDO READ")	
N60 M0	
N70 ENDIF	
...	

1.21 Controle de presença de um arquivo (ISFILE)

Função

Com o comando `ISFILE` é verificada a existência de um arquivo na memória de usuário estática do NCK (sistema passivo de arquivos).

Sintaxe

```
<resultado>=ISFILE("<nome de arquivo>")
```

Significado

`ISFILE`: Comando para controle se o arquivo indicado está presente no sistema passivo de arquivos.

`<nome do arquivo>`: Nome do arquivo cuja presença deve ser controlada no sistema passivo de arquivos.

Tipo: STRING

Ao indicar o nome do arquivo devem ser observados os seguintes itens:

- O nome de arquivo indicado não pode conter espaços vazios ou caracteres de controle (caractere com código ASCII ≤ 32).
- O nome do arquivo pode ser especificado com indicação do caminho e extensão de arquivo:
  - Indicações de caminho

    As indicações de caminho devem ser absolutas, isto é, elas começam com "/".

    Sem a indicação do caminho o arquivo será procurado no atual diretório (=diretório do programa selecionado).
  - Extensão de arquivo

    Se o nome de arquivo não possui nenhum identificador de domínio ("_N_"), ele será complementado de acordo.

    Se no nome de arquivo o quarto último caractere for um sublinhado "_", então os três caracteres seguintes serão interpretados como extensão de arquivo. Para utilizar o mesmo nome de arquivo em todos comandos de arquivo, por exemplo, através de uma variável do tipo STRING, somente podem ser utilizadas as extensões de arquivo _SPF e _MPF.

    Se nenhuma extensão "_MPF" ou "_SPF" for especificada, é complementado automaticamente com _MPF.
- O nome do arquivo não pode ter mais de 32 bytes, o tamanho da indicação do caminho não pode ter mais de 128 bytes.

Exemplo:

```
"PROTFILE"
"_N_PROTFILE"
"_N_PROTFILE_MPF"
"/_N_MPF_DIR_/_N_PROTFILE_MPF/"
```

<resultado>: Variável de resultado para receber o resultado do controle
Tipo. BOOL
Valor: TRUE Arquivo presente
FALSE Arquivo não presente

Exemplo

Código de programa	Comentário
N10 DEF BOOL RESULT	; Definição das variáveis de resultado.
N20 RESULT=ISFILE("TESTFILE")	
N30 IF(RESULT==FALSE)	
N40 MSG("ARQUIVO INEXISTENTE")	
N50 M0	
N60 ENDIF	
...	

ou:

Código de programa	Comentário
N10 DEF BOOL RESULT	; Definição das variáveis de resultado.
N20 RESULT=ISFILE("TESTFILE")	
N30 IF(NOT ISFILE("TESTFILE"))	
N40 MSG("ARQUIVO INEXISTENTE")	
N50 M0	
N60 ENDIF	
...	

1.22 Leitura das informações de arquivo (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO)

Função

Através dos comandos FILEDATE, FILETIME, FILESIZE, FILESTAT e FILEINFO é possível ler (extrair) determinadas informações de arquivo como data / horário do último acesso de gravação, atual tamanho de arquivo, estado de arquivo ou a soma destas informações.

Indicação

O arquivo precisa estar na memória de usuário estática do NCK (sistema de arquivos passivo).

Pré-requisito

O atual nível de proteção ajustado deve ser igual ou mais alto do que o direito Show do diretório de um nível acima. Se este não for o caso, o acesso será negado com mensagem de erro (valor retornado das variáveis com erro = 13).

Sintaxe

```
DEF INT <erro>
DEF STRING[<tamanho de String>] <resultado>
FILE....(<erro>,"<nome de arquivo>",<resultado>)
```

Significado

FILEDATE:	O comando FILEDATE retorna a **data** do último acesso de gravação no arquivo indicado.
FILETIME:	O comando FILETIME retorna o **horário** do último acesso de gravação no arquivo indicado.
FILESIZE:	O comando FILESIZE retorna o **atual tamanho** do arquivo indicado.
FILESTAT:	O comando FILESTAT retorna o **estado** de direitos de leitura, gravação e execução do arquivo indicado.
FILEINFO:	O comando FILEINFO retorna a **soma das informações de arquivo** do arquivo indicado, que podem ser lidas (retornadas) através do FILEDATE, FILETIME, FILESIZE e FILESTAT.

`<erro>`: Variável para o retorno do valor de erro (parâmetro Call-By-Reference)

Tip. INT

Valor:		
	0	nenhum erro
	1	Caminho não permitido
	2	Caminho não encontrado
	3	Arquivo não encontrado
	4	Tipo de arquivo incorreto
	13	Direito de acesso insuficiente
	22	O tamanho de String das variáveis de resultado (`<resultado>`) é muito pequeno.

`<nome do arquivo>`: Nome do arquivo de onde devem ser lidas (extraídas) as informações de arquivo.

Tipo: STRING

Ao indicar o nome do arquivo devem ser observados os seguintes itens:

- O nome de arquivo indicado não pode conter espaços vazios ou caracteres de controle (caracteres de código ASCII ≤ 32), caso contrário o comando `FILE...` será cancelado com o identificador de erro 1 "Caminho não permitido".
- O nome do arquivo pode ser especificado com indicação do caminho e extensão de arquivo:
  - Indicações de caminho

    As indicações de caminho devem ser absolutas, isto é, elas começam com "/".

    Sem a indicação do caminho o arquivo será procurado no atual diretório (=diretório do programa selecionado).
  - Extensão de arquivo

    Se o nome de arquivo não possui nenhum identificador de domínio ("_N_"), ele será complementado de acordo.

    Se no nome de arquivo o quarto último caractere for um sublinhado "_", então os três caracteres seguintes serão interpretados como extensão de arquivo. Para utilizar o mesmo nome de arquivo em todos comandos de arquivo, por exemplo, através de uma variável do tipo STRING, somente podem ser utilizadas as extensões de arquivo _SPF e _MPF.

    Se nenhuma extensão "_MPF" ou "_SPF" for especificada, é complementado automaticamente com _MPF.
- O nome do arquivo não pode ter mais de 32 bytes, o tamanho da indicação do caminho não pode ter mais de 128 bytes.

Exemplo:

```
"PROTFILE"
"_N_PROTFILE"
"_N_PROTFILE_MPF"
"/_N_MPF_DIR_/_N_PROTFILE_MPF/"
```

`<resultado>`: Variável de resultado (parâmetro Call-By-Reference)

Variável onde a informação de arquivo solicitada é armazenada.

Tipo: STRING para: `FILEDATE`
Formato: "dd.mm.yy"
⇒ O tamanho da String deve ser **8**.
`FILETIME`
Formato: " hh:mm.ss "
⇒ O tamanho da String deve ser **8**.
`FILESTAT`
Formato: "rwxsd"
(r: **r**ead, w: **w**rite, x: e**x**ecute, s: **s**how, d: **d**elete)
⇒ O tamanho da String deve ser **5**.
`FILEINFO`
Formato: "rwxsd nnnnnnnn dd.mm.yy hh:mm:ss"
⇒ O tamanho da String deve ser **32**.

INT para: `FILESIZE`
O tamanho de arquivo é retornado em Bytes.

Exemplo

Código de programa	**Comentário**
`N10 DEF INT ERROR`	`; Definição das variáveis de erro.`
`N20 STRING[32] RESULT`	`; Definição das variáveis de resultado.`
`N30 FILEINFO(ERROR,"/_N_MPF_DIR/_N_TESTFILE_MPF",RESULT)`	`; Nome de arquivo com identificadores de domínio e de arquivo e indicação do caminho.`
`N40 IF ERROR <>0`	`; Avaliação de erro`
`N50 MSG("ERRO"<<ERROR<<"COM COMANDO FILEINFO")`	
`N60 M0`	
`N70 ENDIF`	
`...`	

Por exemplo, o exemplo poderia retornar o seguinte resultado na variável RESULT:

"77777 12345678 26.05.00 13:51:30"

1.23 Cálculo de checksum através de um campo (CHECKSUM)

Função

Com o comando CHECKSUM a somatória de controle (checksum) pode ser calculada através de um campo. Através da comparação destes checksums com o resultado de um cálculo de checksum anterior pode-se verificar se os dados do campo sofreram uma alteração.

Aplicação

Verificação se o contorno de entrada sofreu alteração durante o desbaste.

Sintaxe

```
DEF INT <erro>
DEF STRING[<tamanho de String>] <checksums>
DEF ... <campo>[<n>,<m>,<o>]
<erro>=CHECKSUM(<checksums>,"<campo>"[,<coluna inicial>,<coluna
final>])
```

Significado

CHECKSUM: Comando para calcular os checksums através de um campo

<erro>: Variável para o retorno do valor do erro

Tipo. INT

Valor:

0	nenhum erro
1	Símbolo não encontrado
2	Nenhum campo
3	Índice 1 muito grande
4	Índice 2 muito grande
5	Tipo inválido de dado
10	Checksum excedido

<checksums>: Variável de resultado para receber o resultado do cálculo de checksum (parâmetro Call-By-Reference)

Tipo: STRING

Tamanho de String necessário: 16

O checksum é apresentado como uma seqüência de caracteres de 16 números hexadecimais. Mas não se especifica nenhum caractere de formato.

Exemplo: "A6FC3404E534047C"

<campo>:	Nome do campo através do qual é formado o checksum (parâmetro Call-By-Value) Tipo: STRING Tamanho de String máx.: 32 Os campos permitidos são campos de 1 até 3 dimensões dos tipos: BOOL, CHAR, INT, REAL, STRING **Nota:** Os campos de dados de máquina não são permitidos.
<coluna inicial>:	Número da coluna inicial do campo destinado para o cálculo do checksum (parâmetro opcional)
<coluna final>:	Número da coluna final do campo destinado para o cálculo do checksum (parâmetro opcional)

Indicação

Os parâmetros <coluna inicial> e <coluna final> são opcionais. Se não for especificado nenhum índice de coluna, então o checksum será formado sobre o campo inteiro.

O resultado do checksum sempre é único. Em caso de alteração de um elemento de campo, também resultará outra String de resultado.

Exemplo

```
Código de programa                                  Comentário
N10 DEF INT ERROR                                   ; Definição das variáveis de
                                                      erro.
N20 DEF STRING[16] MY_CHECKSUM                      ; Definição das variáveis de
                                                      resultado.
N30 DEF INT MY_VAR[4,4]                             ; Definição de campos.
N40 MY_VAR=...
N50 ERROR=CHECKSUM(MY_CHECKSUM,"MY_VAR",0,2)
...
```

Por exemplo, o exemplo poderia retornar o seguinte resultado na variável MY_CHECKSUM:

"A6FC3404E534047C"

1.24 Arredondamento (ROUNDUP)

Função

Com a função "ROUNDUP" se pode arredondar os valores de entrada de tipo REAL (números fracionados com ponto decimal) para o próximo número inteiro maior.

Sintaxe

```
ROUNDUP(<valor>)
```

Significado

`ROUNDUP`:	Comando para arredondar um valor de entrada
`<valor>`:	Valor de entrada de tipo REAL

Indicação

Valores de entrada de tipo INTEGER (um número inteiro) são retornados sem alteração.

Exemplos

Exemplo 1: Diversos valores de entrada e seus resultados de arredondamento

Exemplo	Resultado de arredondamento
`ROUNDUP(3.1)`	4.0
`ROUNDUP(3.6)`	4.0
`ROUNDUP(-3.1)`	-3.0
`ROUNDUP(-3.6)`	-3.0
`ROUNDUP(3.0)`	3.0
`ROUNDUP(3)`	3.0

Exemplo 2: ROUNDUP no programa NC

Código de programa

```
N10 X=ROUNDUP(3.5) Y=ROUNDUP(R2+2)
N15 R2=ROUNDUP($AA_IM[Y])
N20 WHEN X=100 DO Y=ROUNDUP($AA_IM[X])
...
```

1.25 Técnica de subrotinas

1.25.1 Informações gerais

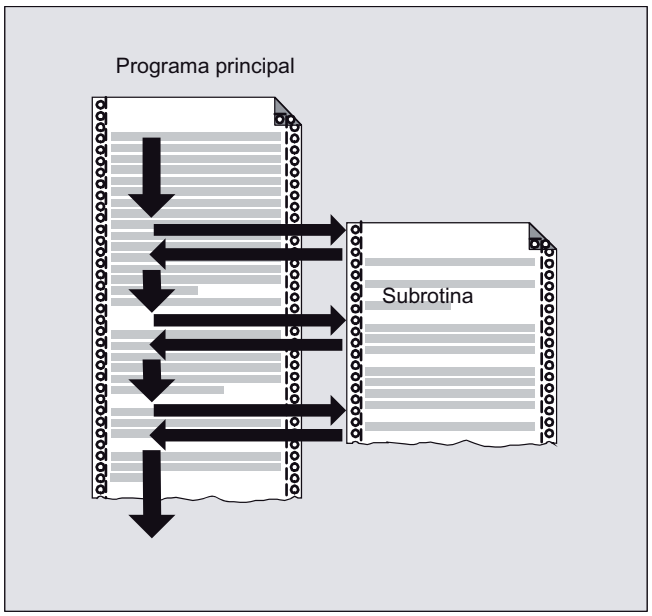
1.25.1.1 Subrotina

Função

A denominação "subrotina" é uma herança do tempo em que se dividia os programas de peças em programas principais e subrotinas. No caso, os programas principais eram os programas de peça que eram selecionados no comando para execução e depois era dada a partida dos mesmos. As subrotinas eram os programas de peça que eram chamados a partir do programa principal.

Esta divisão não existe mais na atual linguagem NC do SINUMERIK. Cada programa de peça pode, em princípio, ser selecionado e iniciado como programa principal, ou então ser chamado como subrotina a partir de outro programa de peça.

Com isso, nos demais procedimentos, uma subrotina é denominada como um programa de peça que pode ser chamado a partir de outro programa de peça.



Aplicação

Como em todas as linguagens de programação de nível mais alto, na linguagem NC as subrotinas também são aplicadas para armazenar (comportar) programas fechados e independentes, que utilizam partes de programa múltiplas vezes.

As subrotinas oferecem as seguintes vantagens:

- Melhoram a clareza e leitura dos programas
- Melhoram a qualidade através do reaproveitamento de partes de programa já testados
- Oferecem a opção para aquisição de bibliotecas específicas de usinagem
- Economizam espaço na memória

1.25.1.2 Nomes de subrotina

Regras para denominação

Para a atribuição de nomes de subrotinas devem ser observadas as seguintes regras:

- Os dois primeiros caracteres devem ser letras (A - Z, a - z).
- Os caracteres seguintes podem ser utilizados em qualquer combinação de letras, números (0 - 9) e sublinhados ("_").
- É permitido o uso de até 31 caracteres.

Indicação

Na linguagem NC do SINUMERIK **não** é feita nenhuma distinção entre letras maiúsculas e letras minúsculas.

Ampliações do nome do programa

O nome de programa atribuído na criação do programa é ampliado internamente no comando com um prefixo e um sufixo:

- Prefixo: _N_
- Sufixo:
  - Programas principais: _MPF
  - Subrotinas: _SPF

Utilização do nome de programa

Para a utilização do nome de programa, por exemplo em uma chamada de subrotina, são possíveis todas as combinações de prefixo, nome de programa e sufixo.

Exemplo:

A subrotina com o nome de programa "SUB_PROG" pode ser iniciada através das seguintes chamadas:

1. `SUB_PROG`
2. `_N_SUB_PROG`
3. `SUB_PROG_SPF`
4. `_N_SUB_PROG_SPF`

Indicação

Igualdade de nome de programas principais e subrotinas

Programas principais (.MPF) e subrotinas (.SPF) que existirem com o mesmo nome de programa, devem receber o sufixo correspondente quando os nomes de programas forem utilizados no programa de peça, para identificar de forma clara cada programa.

1.25.1.3 Aninhamento de subrotinas

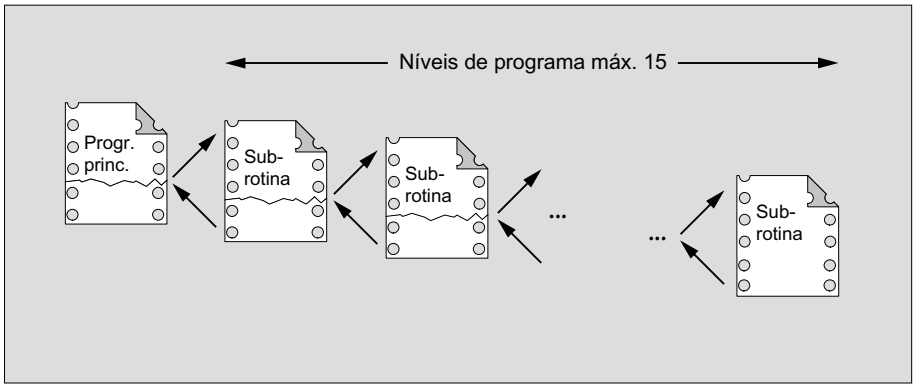
Um programa principal pode chamar subrotinas, que por sua vez podem chamar outras subrotinas. Dessa forma as execuções dos programas estão agrupadas uma dentro da outra. Por isso que cada programa é processado em um nível de programa próprio.

Nível de agrupamento

Atualmente a linguagem NC coloca 16 níveis de programa à disposição. O programa principal sempre é processado no nível de programa mais alto, o nível 0. Uma subrotina sempre é processada no próximo nível de programa mais baixo a partir da chamada. Assim o nível de programa 1 é o primeiro nível de subrotina.

Divisão dos níveis de programa:

- Nível de programa 0: Nível de programa principal
- Nível de programa 1 - 15: Nível de subrotina 1 - 15



Rotinas de interrupção (ASUP)

Se uma subrotina é chamada durante uma rotina de interrupção, esta não será processada no atual nível de programa (n) ativo no canal, mas no próximo nível de programa mais baixo (n+1). Para que isto também seja possível no nível de programa mais baixo, estão disponíveis 2 níveis de programa extras (16 e 17) quando relacionados com rotinas de interrupção.

Se forem necessários mais que 2 níveis de programa, isto deve ser considerado de modo explícito na estruturação do programa de peça executado no canal. Isto significa que somente pode ser utilizado um número de níveis de programa que ainda proporcione a quantidade de níveis de programa suficiente para a execução da interrupção.

Por exemplo, se a execução da interrupção precisar de 4 níveis de programa, o programa de peça deve ser estruturado de modo que ele ocupe no máximo até o nível de programa 13. Quando ocorre uma interrupção, estes 4 níveis de programa (14 até 17) necessários estarão disponíveis.

Ciclos da Siemens

Os ciclos da Siemens requerem 3 níveis de programa. Por isso que a chamada de um ciclo da Siemens deve ocorrer no máximo até:

- Na execução do programa de peça: Nível de programa 12
- Na rotina de interrupção: Nível de programa 14

1.25.1.4 Caminho de localização

Na chamada de uma subrotina sem indicação do caminho o comando procura na seguinte ordem de diretórios indicada:

Seqüência	Diretório	Descrição
1.	Atual diretório	Diretório do programa de chamada
2.	/_N_SPF_DIR /	Diretório de subrotinas global
3.	/_N_CUS_DIR /	Ciclos do usuário
4.	/_N_CMA_DIR /	Ciclos do fabricante
5.	/_N_CST_DIR /	Ciclos padronizados

1.25.1.5 Parâmetros formais e parâmetros atuais

Os parâmetros formais e parâmetros atuais estão relacionados com a definição e a chamada de subrotinas com transferência de parâmetros.

Parâmetro formal

Durante a definição de uma subrotina os parâmetros que devem ser transferidos à subrotina, os chamados parâmetros formais, devem ser definidos com tipo e nome de parâmetro.

Com isso os parâmetros formais definem a interface da subrotina.

Exemplo:

```
Código de programa                      Comentário
PROC CONTORNO (REAL X, REAL Y)          ; Parâmetro formal: X e Y ambos do tipo REAL
N20 X1=X Y1=Y                           ; Deslocamento do eixo X1 até a posição X e do
                                          eixo Y1 até a posição Y
...
N100 RET
```

Parâmetro atual

Durante a chamada de uma subrotina devem ser transferidos à subrotina valores absolutos ou variáveis, chamados de parâmetros atuais.

Com isso os parâmetros atuais alimentam os valores atuais durante a chamada da interface da subrotina.

Exemplo:

```
Código de programa                      Comentário
N10 DEF REAL LARGURA                    ; Definição de variáveis
N20 LARGURA=20.0                        ; Atribuição de variáveis
N30 CONTORNO(5.5, LARGURA)              ; Chamada de subrotina com parâmetros atuais:
                                          5.5 e LARGURA
...
N100 M30
```

1.25.1.6 Transferência de parâmetros

Definição de uma subrotina com transferência de parâmetros

A definição de uma subrotina com transferência de parâmetros é realizada com a palavra-chave `PROC` e uma listagem completa de todos os parâmetros esperados pela subrotina.

Transferência incompleta de parâmetros

Na chamada da subrotina nem sempre precisam ser transferidos de modo explícito todos os parâmetros definidos na interface da subrotina. Quando um parâmetro é desconsiderado, para este parâmetro é transferido o valor padrão "0".

Entretanto, para uma identificação única da ordem dos parâmetros sempre devem ser indicadas as vírgulas como caracteres de separação dos parâmetros. Uma exceção é o último parâmetro. Se este não for considerado na chamada, também pode ser descartada a última vírgula.

Exemplo:

Subrotina:

```
Código de programa                              Comentário
PROC SUB_PROG (REAL X, REAL Y, REAL Z)          ;  Parâmetro formal: X, Y e Z
...
N100 RET
```

Programa principal:

```
Código de programa                  Comentário
PROC MAIN_PROG
...
N30 SUB_PROG(1.0,2.0,3.0)           ; Chamada de subrotina com transferência de
                                      parâmetros completa:
                                      X=1.0, Y=2.0, Z=3.0
…
N100 M30
```

Exemplos para a chamada de subrotina no `N30` com transferência de parâmetros incompleta:

```
N30 SUB_PROG( ,2.0,3.0)          ; X=0.0, Y=2.0, Z=3.0
N30 SUB_PROG(1.0, ,3.0)          ; X=1.0, Y=0.0, Z=3.0
N30 SUB_PROG(1.0,2.0)            ; X=1.0, Y=2.0, Z=0.0
N30 SUB_PROG( , ,3.0)            ; X=0.0, Y=0.0, Z=3.0
N30 SUB_PROG( , , )              ; X=0.0, Y=0.0, Z=0.0
```

CUIDADO

Transferência de parâmetros Call-by-Reference

Os parâmetros que são transferidos através de Call-by-Reference, não podem ser desconsiderados na chamada de subrotina.

CUIDADO

Tipo de dados AXIS

Os parâmetros do tipo de dados AXIS não podem ser desconsiderados na chamada de subrotina.

Verificação dos parâmetros de transferência

Através da variável de sistema $P_SUBPAR [n] mit n = 1, 2, ... pode ser verificado na subrotina se um parâmetro foi transferido de modo explícito ou se ele foi desconsiderado. O índice n refere-se à ordem dos parâmetros formais. O índice n = 1 refere-se ao 1º parâmetro formal, o índice n = 2 ao 2º parâmetro formal, e assim por diante.

O seguinte segmento de programa mostra no exemplo do 1º parâmetro formal, como que uma verificação pode ser realizada:

```
Programação                                  Comentário
PROC SUB_PROG (REAL X, REAL Y, REAL Z)       ; Parâmetro formal: X, Y e Z
N20 IF $P_SUBPAR[1]==TRUE                    ; Verificação do 1° parâmetro formal X.
...                                          ; Estas ações são executadas quando o
                                               parâmetro formal X foi transferido
                                               explicitamente.
N40 ELSE
...                                          ; Estas ações são executadas quando o
                                               parâmetro formal X não foi
                                               transferido.
N60 ENDIF
...                                          ; Ações gerais
N100 RET
```

1.25.2 Definição de uma subrotina

1.25.2.1 Subrotina sem transferência de parâmetros

Função

Para a definição de subrotinas sem transferência de parâmetros pode ser descartada a linha de definição no início do programa.

Sintaxe

```
[PROC <nome de programa>]
...
```

Significado

`PROC`:	Instrução de definição no início de um programa
`<nome de programa>`:	Nome do programa

Exemplo

Exemplo 1: Subrotina com instrução `PROC`

Código de programa	**Comentário**
`PROC SUB_PROG`	`; Linha de definição`
`N10 G01 G90 G64 F1000`	
`N20 X10 Y20`	
`...`	
`N100 RET`	`; Salto de retorno da subrotina`

Exemplo 2: Subrotina sem instrução `PROC`

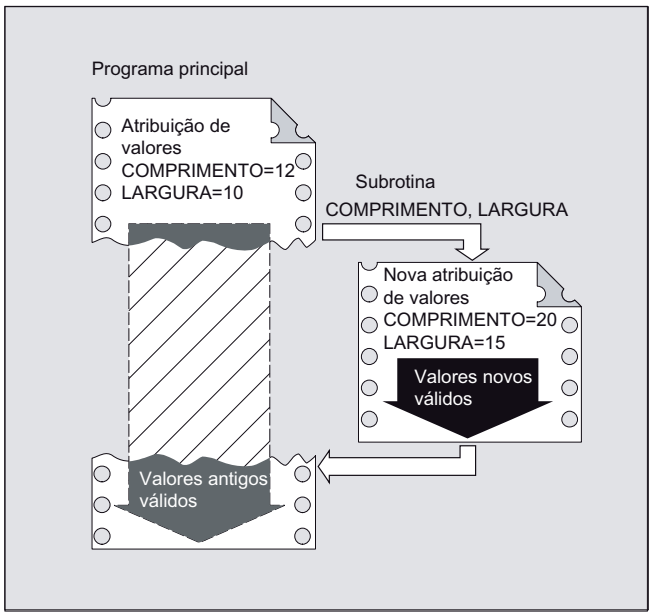
Código de programa	**Comentário**
`N10 G01 G90 G64 F1000`	
`N20 X10 Y20`	
`...`	
`N100 RET`	`; Salto de retorno da subrotina`

1.25.2.2 Subrotina com transferência de parâmetros Call-by-Value (PROC)

Função

A definição de uma subrotina com transferência de parâmetros Call-By-Value é realizada com a palavra-chave `PROC` seguida de nomes de programa e uma listagem completa de todos os parâmetros esperados pela subrotina, com indicação de tipo e nome. A instrução de definição deve estar na primeira linha do programa.

A transferência de parâmetros Call-by-Value não tem nenhuma reação sobre o programa de chamada. O programa de chamada transfere à subrotina apenas os valores dos parâmetros atuais.



Indicação

Podem ser transferidos no máximo 127 parâmetros.

Sintaxe

```
PROC <nome de programa> (<tipo de parâmetro> <nome de parâmetro>,
...)
```

Significado

`PROC`:	Instrução de definição no início de um programa
`<nome de programa>`:	Nome do programa
`<tipo de parâmetro>`:	Tipo de dados do parâmetro (p. ex. REAL, INT, BOOL)
`<nome de parâmetro>`:	Nome do parâmetro

ATENÇÃO
O nome de programa informado após a palavra-chave PROC deve coincidir com o nome de programa dado na interface de operação.

Exemplo

Definição de uma subrotina com 2 parâmetros do tipo REAL:

Código de programa	Comentário
PROC SUB_PROG (REAL COMPRIMENTO, REAL LARGURA)	; Parâmetro 1: Tipo: REAL, nome: COMPRIMENTO Parâmetro 2: Tipo: REAL, nome: LARGURA
...	
N100 RET	; Salto de retorno da subrotina

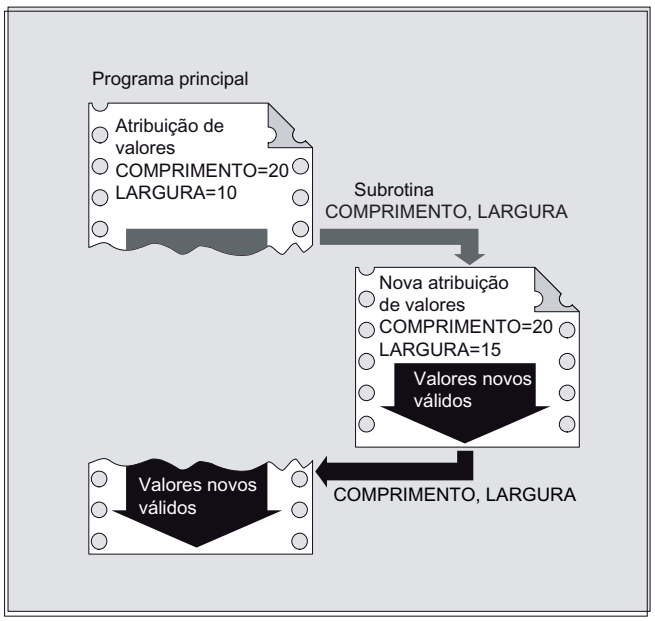
1.25.2.3 Subrotina com transferência de parâmetros Call-by-Reference (PROC, VAR)

Função

A definição de uma subrotina com transferência de parâmetros Call-By-Reference é realizada com a palavra-chave `PROC` seguida de nomes de programa e uma listagem completa de todos os parâmetros esperados pela subrotina, com indicação `VAR`de tipo e nome. A instrução de definição deve estar na primeira linha do programa.

Na transferência de parâmetros Call-By-Reference também podem ser transferidas referências para os campos.

A transferência de parâmetros Call-by-Reference tem reações sobre o programa de chamada. O programa de chamada transfere à subrotina uma referência sobre o parâmetro atual, permitindo assim à subrotina um acesso direto à variável correspondente.



Indicação

Podem ser transferidos no máximo 127 parâmetros.

Indicação

Uma transferência de parâmetros Call-by-Reference somente é necessária se a variável de transferência for definida no programa de chamada (LUD). As variáveis globais de canal ou as variáveis globais do NC não precisam ser transferidas, pois estas podem ser acessadas diretamente pela subrotina.

Sintaxe

```
PROC <nome de programa> (VAR <tipo de parâmetro> <npme de
parâmetro>, ...)
PROC <nome de programa> (VAR <tipo de campo> <nome de campo>
[<m>,<n>,<o>], ...)
```

Significado

`PROC`:	Instrução de definição no início de um programa
`VAR`:	Palavra-chave para a transferência de parâmetros por referência
`<nome de programa>`:	Nome do programa
`<tipo de parâmetro>`:	Tipo de dados do parâmetro (p. ex. REAL, INT, BOOL)
`<nome de parâmetro>`:	Nome do parâmetro
`<tipo de campo>`:	Tipo de dados dos elementos de campo (p. ex. REAL, INT, BOOL)
`<nome de campo>`:	Nome do campo
`[<m>,<n>,<o>]`:	Tamanho de campo
	Atualmente, no máximo são possíveis campos de 3 dimensões:
	`<m>`: Tamanho de campo para 1ª dimensão
	`<n>`: Tamanho de campo para 2ª dimensão
	`<o>`: Tamanho de campo para 3ª dimensão

ATENÇÃO

O nome de programa informado após a palavra-chave `PROC` deve coincidir com o nome de programa dado na interface de operação.

Indicação

Com campos de tamanho indefinido como parâmetros formais se pode processar subrotinas de campos de tamanho variável. Para isso não se especifica, por exemplo, o tamanho da 1ª dimensão no momento da definição de um campo de duas dimensões como parâmetro formal. Porém, a vírgula precisa ser escrita.

Exemplo: `PROC <nome de programa> (VAR REAL CAMPO[ ,5])`

Exemplo

Definição de uma subrotina com 2 parâmetros como referência ao tipo REAL:

Código de programa	**Comentário**
`PROC SUB_PROG(VAR REAL COMPRIMENTO, VAR REAL LARGURA)`	`; Parâmetro 1: Referência ao tipo: REAL, nome: COMPRIMENTO` `Parâmetro 2: Referência ao tipo: REAL, nome: LARGURA`
`...`	
`N100 RET`	

1.25.2.4 Salvar funções G modais (SAVE)

Função

O atributo SAVE faz com que todas funções G modais ativas antes da chamada da subrotina sejam salvas e novamente ativadas após o fim da subrotina.

CUIDADO
Interrupção do modo de controle da trajetória
Se com o modo de controle da trajetória ativo uma subrotina for chamada com o atributo SAVE, o modo de controle da trajetória é interrompido no fim da subrotina (salto de retorno).

Sintaxe

```
PROC <nome da subrotina> SAVE
```

Significado

SAVE: Salvamento das funções G antes da chamada da subrotina e seu restabelecimento após o fim da subrotina

Exemplo

Na subrotina CONTORNO atua a função G modal G91 (dimensão incremental). No programa principal atua a função G modal G90 (dimensão absoluta). Através da definição de subrotina com o SAVE o G90 torna-se novamente ativo no programa principal após o fim da subrotina.

Definição de subrotina:

Código de programa	Comentário
PROC CONTORNO (REAL VALOR1) SAVE	; Definição de subrotina com parâmetro SAVE
N10 G91 ...	; Função G modal G91: Dimensão incremental
N100 M17	; Fim de subrotina

Programa principal:

Código de programa	Comentário
N10 G0 X... Y... G90	; Função G modal G90: Dimensão absoluta
N20 ...	
...	
N50 CONTORNO (12.4)	; Chamada de subrotina
N60 X... Y...	; Função G modal G90 reativada através do SAVE

Condições gerais

Frames

A relação dos Frames em relação às subrotinas com o atributo SAVE depende do tipo do Frame e pode ser ajustada através de dados de máquina.

Literatura

Manual de funções básicas, eixos, sistemas de coordenadas, Frames (K2),
capítulo: "Salto de retorno de subrotinas com SAVE"

1.25.2.5 Supressão de processamento bloco a bloco (SBLOF, SBLON)

Função

Supressão de bloco a bloco para o programa inteiro

Os programas identificados com SBLOF são executados inteiros, mesmo com o processamento bloco a bloco ativo, ou seja, o processamento bloco a bloco é suprimido em todo o programa.

SBLOF está na linha PROC e vale até o fim ou cancelamento da subrotina. Com o comando de salto de retorno se decide se a parada deve ser feita no fim da subrotina ou não:

Salto de retorno com M17:	Parada no fim da subrotina
Salto de retorno com RET:	Nenhuma parada no fim da subrotina

Supressão de bloco a bloco dentro do programa

O SBLOF deve constar isolado em um bloco. A partir deste bloco se desativa o bloco a bloco até:

- o próximo SBLON ou

  ou

- o fim do nível ativo da subrotina

Sintaxe

Supressão de bloco a bloco para o programa inteiro:

```
PROC ... SBLOF
```

Supressão de bloco a bloco dentro do programa:

```
SBLOF
...
SBLON
```

Significado

PROC:	Primeira instrução de um programa
SBLOF:	Comando para desativação do processamento bloco a bloco O SBLOF pode estar em um bloco PROC ou isolado em um bloco.
SBLON:	Comando para ativar o processamento bloco a bloco O SBLON deve estar em um bloco próprio.

Condições gerais

- **Supressão de bloco a bloco e exibição de bloco**

  A atual exibição de bloco pode ser ocultada em ciclos/subrotinas com DISPLOF. Se o DISPLOF for programado junto com o SBLOF, então será exibida a chamada do ciclo/subrotina na parada do bloco a bloco durante o ciclo/subrotina.

- **Supressão de bloco a bloco na ASUP de sistema na ASUP de usuário**

  Se a parada de bloco a bloco na ASUP de sistema ou de usuário for suprimida através das configurações no dado de máquina MD10702 $MN_IGNORE_SINGLEBLOCK_MASK (Bit0 = 1 e Bit1 = 1), então a parada de bloco a bloco pode ser ativada novamente através da programação do SBLON na ASUP.

  Se a parada de bloco a bloco for suprimida na ASUP de usuário através da configuração no dado de máquina MD20117 $MC_IGNORE_SINGLEBLOCK_ASUP, então a parada de bloco a bloco **não** pode ser ativada novamente através da programação de SBLON na ASUP.

- **Particularidades da supressão de bloco a bloco nos diversos tipos de processamento bloco a bloco**

  Com o processamento bloco a bloco SBL2 (parada após cada bloco de programa de peça) ativo **não** se executa a parada no bloco do SBLON se no MD10702 $MN_IGNORE_SINGLEBLOCK_MASK (evitar parada de bloco a bloco) o Bit 12 estiver ajustado em "1".

  Com o processamento bloco a bloco SBL3 (parada após cada bloco de programa de peça e também no ciclo) se suprimir o comando SBLOF.

Exemplos

Exemplo 1: Supressão de bloco a bloco dentro de um programa

```
Código de programa          Comentário
N10 G1 X100 F1000
N20 SBLOF                   ; Desativar bloco a bloco
N30 Y20
N40 M100
N50 R10=90
N60 SBLON                   ; Ativar novamente bloco a bloco
N70 M110
N80 ...
```

A área entre N20 e N60 é processada como um passo em modo bloco a bloco.

Exemplo 2: O ciclo deve atuar como um comando para o usuário

Programa principal:

```
Código de programa
N10 G1 X10 G90 F200
N20 X-4 Y6
N30 CYCLE1
N40 G1 X0
N50 M30
```

Ciclo CYCLE1:

```
Código de programa                          Comentário
N100 PROC CYCLE1 DISPLOF SBLOF              ; Suprimir bloco a bloco
N110 R10=3*SIN(R20)+5
N120 IF (R11 <= 0)
N130 SETAL(61000)
N140 ENDIF
N150 G1 G91 Z=R10 F=R11
N160 M17
```

O ciclo CYCLE1 é executado com processamento bloco a bloco ativo, isto é, deve-se pressionar uma vez a tecla Start para execução do CYCLE1.

Exemplo 3:

Uma ASUP iniciada pelo PLC para a ativação do deslocamento modificado do ponto zero e para correções da ferramenta não deverá estar visível.

```
Código de programa
N100 PROC NV SBLOF DISPLOF
N110 CASE $P_UIFRNUM OF     0 GOTOF _G500
                            1 GOTOF _G54
                            2 GOTOF _G55
                            3 GOTOF _G56
                            4 GOTOF _G57
                            DEFAULT GOTOF END
N120 _G54: G54 D=$P_TOOL T=$P_TOOLNO
N130 RET
N140 _G54: G55 D=$P_TOOL T=$P_TOOLNO
N150 RET
N160 _G56: G56 D=$P_TOOL T=$P_TOOLNO
N170 RET
N180 _G57: G57 D=$P_TOOL T=$P_TOOLNO
N190 RET
N200 END: D=$P_TOOL T=$P_TOOLNO
N210 RET
```

Exemplo 4: Com MD10702 Bit 12 = 1 são se executa a parada

Situação inicial:

- O processamento bloco a bloco está ativo.
- MD10702 $MN_IGNORE_SINGLEBLOCK_MASK Bit12 = 1

Programa principal:

Código de programa	Comentário
N10 G0 X0	; Parar nesta linha do programa de peça.
N20 X10	; Parar nesta linha do programa de peça.
N30 CYCLE	; Bloco de deslocamento gerado pelo ciclo.
N50 G90 X20	; Parar nesta linha do programa de peça.
M30	

Ciclo CYCLE:

Código de programa	Comentário
PROC CYCLE SBLOF	; Suprimir parada do bloco a bloco
N100 R0 = 1	
N110 SBLON	; Por causa do MD10702 Bit12=1 não se executa a parada nesta linha do programa de peça.
N120 X1	; Se executa a parada nesta linha do programa de peça.
N140 SBLOF	
N150 R0 = 2	
RET	

Exemplo 5: Supressão de bloco a bloco com aninhamento de programas

Situação inicial:

O processamento bloco a bloco está ativo.

Aninhamento de programas:

```
Código de programa                          Comentário
N10 X0 F1000                                ; Neste bloco se executa a parada.
N20 UP1(0)
      PROC UP1(INT _NR) SBLOF               ; Suprimir parada do bloco a bloco.
      N100 X10
      N110 UP2(0)
            PROC UP2(INT _NR)
            N200 X20
            N210 SBLON                      ; Ativar parada do bloco a bloco.
            N220 X22                        ; Neste bloco se executa a parada.
            N230 UP3(0)
                  PROC UP3(INT _NR)
                  N300 SBLOF                ; Suprimir parada do bloco a bloco.
                  N305 X30
                  N310 SBLON                ; Ativar parada do bloco a bloco.
                  N320 X32                  ; Neste bloco se executa a parada.
                  N330 SBLOF                ; Suprimir parada do bloco a bloco.
                  N340 X34
                  N350 M17                  ; SBLOF está ativo.
            N240 X24                        ; Neste bloco se executa a parada. SBLON
                                              está ativo.
            N250 M17                        ; Neste bloco se executa a parada. SBLON
                                              está ativo.
      N120 X12
      N130 M17                              ; Neste bloco de salto de retorno se
                                              executa a parada. O SBLOF da instrução
                                              PROC está ativo.
N30 X0                                      ; Neste bloco se executa a parada.
N40 M30                                     ; Neste bloco se executa a parada.
```

Outras informações

Bloqueio de bloco a bloco para subrotinas assíncronas

Para executar uma ASUP em um passo no modo bloco a bloco, na ASUP deve ser programada uma instrução PROC com SBLOF. Isto também se aplica para a função "ASUP editável de sistema" (MD11610 $MN_ASUP_EDITABLE).

Exemplo para uma ASUP editável de sistema:

```
Código de programa                          Comentário
N10 PROC ASUP1 SBLOF DISPLOF
N20 IF $AC_ASUP=='H200'
N30 RET                                     ; Sem REPOS na mudança de modo de operação.
N40 ELSE
N50 REPOSA                                  ; REPOS nos demais casos.
N60 ENDIF
```

Controles de programa em bloco a bloco

No processamento bloco a bloco o usuário pode executar o programa de peça bloco a bloco. Existem os seguintes tipos de ajuste:

- SBL1: Bloco a bloco IPO com parada após cada bloco de função da máquina.
- SBL2: Bloco a bloco com parada após cada bloco.
- SBL3: Parada no ciclo (com a seleção do SBL3 se suprime o comando SBLOF).

Supressão de bloco a bloco com aninhamento de programas

Se em uma subrotina foi programado SBLOF na instrução PROC, então a parada é feita com M17 no salto de retorno da subrotina. Com isso evita-se que seja executado o próximo bloco do programa de chamada. Se em uma subrotina com SBLOF for ativada uma supressão de bloco a bloco sem SBLOF na instrução PROC, então a parada somente será realizada após o próximo bloco de função da máquina do programa chamado. Se isto não for desejado, então na subrotina se deve programar novamente o SBLON, ainda antes do retorno (M17). A parada não será realizada no caso de um salto de retorno com RET para um programa de nível superior.

1.25.2.6 Supressão da atual exibição de bloco (DISPLOF, DISPLON, ACTBLOCNO)

Função

Como padrão, na exibição de bloco se exibe o atual bloco de programa. A exibição do atual bloco pode ser ocultada nos ciclos e nas subrotinas com o comando DISPLOF. Ao invés do atual bloco se exibe a chamada do ciclo ou da subrotina. Com o comando DISPLON cancela-se novamente a supressão da exibição de bloco.

O DISPLOF e o DISPLON é programado na linha de programa com a instrução PROC e tem efeito sobre toda a subrotina e de forma implícita para todas as subrotinas chamadas por esta subrotina, que não tiverem o comando DISPLON ou DISPLOF. Este comportamente também é válido para ASUPs.

Sintaxe

```
PROC … DISPLOF
PROC … DISPLOF ACTBLOCNO
PROC … DISPLON
```

Significado

DISPLOF: Comando para ocultar a atual exibição de bloco.

Posicionamento: No fim da linha de programa com a instrução PROC

Efeito: Até o salto de retorno da subrotina ou fim de programa.

Nota:
Se mais subrotinas forem chamadas a partir da subrotina com o comando DISPLOF, então nesta também se oculta a atual exibição de bloco, isto se nesta não for programado explicitamente o DISPLON.

DISPLON: Comando para cancelar a supressão da atual exibição de bloco

Posicionamento: No fim da linha de programa com a instrução PROC

Efeito: Até o salto de retorno da subrotina ou fim de programa.

Nota:
Se mais subrotinas forem chamadas a partir da subrotina com o comando DISPLON, então nesta também se exibe o atual bloco de programa, isto se neste não for programado explicitamente o DISPLOF.

ACTBLOCNO: O DISPLOF junto com o atributo ACTBLOCNO faz com que, em caso de um alarme, seja retornado o número do atual bloco em que ocorreu o alarme. Isto também se aplica se em um nível de programa mais baixo apenas estiver programado o DISPLOF.

Em contrapartida, com DISPLOF sem ACTBLOCNO, o número do bloco da chamada do ciclo ou da subrotina não é exibido no último nível de programa marcado com DISPLOF.

Exemplos

Exemplo 1: Ocultar a atual exibição de bloco no ciclo

Código de programa	Comentário
`PROC CYCLE(AXIS TOMOV, REAL POSITION) SAVE DISPLOF`	`; Ocultar atual exibição de bloco. O invés disso, deve ser exibida a chamada do ciclo, p. ex.: CYCLE(X,100.0)`
`DEF REAL DIFF`	`; Conteúdo do ciclo`
`G01 ...`	
`...`	
`RET`	`; Salto de retorno da subrotina. Na exibição de bloco se exibe o bloco seguinte à chamada do ciclo.`

Exemplo 2: Exibição de bloco na emissão de alarme

Subrotina SUBPROG1 (com ACTBLOCNO):

Código de programa	Comentário
`PROC SUBPROG1 DISPLOF ACTBLOCNO`	
`N8000 R10 = R33 + R44`	
`...`	
`N9040 R10 = 66 X100`	`; Disparar alarme 12080`
`...`	
`N10000 M17`	

Subrotina SUBPROG2 (sem ACTBLOCNO):

Código de programa	Comentário
`PROC SUBPROG2 DISPLOF`	
`N5000 R10 = R33 + R44`	
`...`	
`N6040 R10 = 66 X100`	`; Disparar alarme 12080`
`...`	
`N7000 M17`	

Programa principal:

```
Código de programa          Comentário
N1000 G0 X0 Y0 Z0
N1010 ...
...
N2050 SUBPROG1              ; Emissão de alarme = "12080 Canal K1 Bloco N9040
                              Erro de sintaxe no texto R10="
N2060 ...
N2350 SUBPROG2              ; Emissão de alarme = "12080 Canal K1 Bloco N2350
                              Erro de sintaxe no texto R10="
...
N3000 M30
```

Exemplo 3: Cancelamento da supressão da atual exibição de bloco

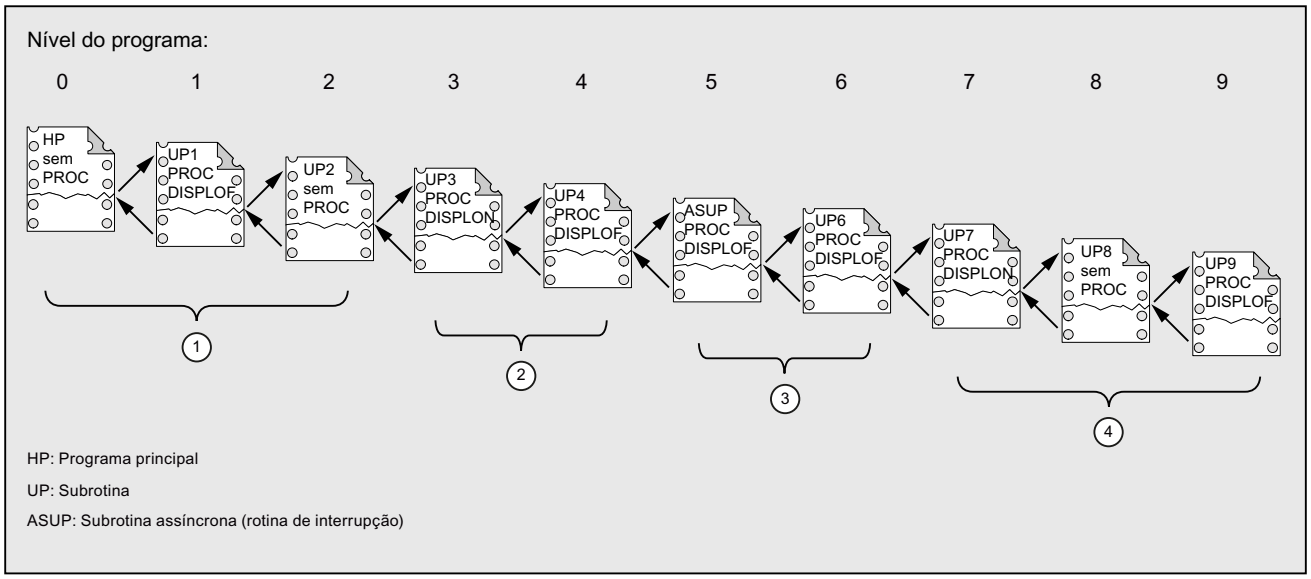
Subrotina SUB1 com supressão:

```
Código de programa          Comentário
PROC SUB1 DISPLOF           ; Suprime a atual exibição de bloco na subrotina
                              SUB1. Ao invés disso, o bloco deve ser exibido com
                              a chamada da SUB1.
...
N300 SUB2                   ; Chama a subrotina SUB2.
...
N500 M17
```

Subrotina SUB1 sem supressão:

```
Código de programa          Comentário
PROC SUB2 DISPLON           ; Cancela a supressão da atual exibição de bloco na
                              subrotina SUB2.
...
N200 M17                    ; Salto de retorno para a subrotina SUB1. Na SUB1 é
                              novamente sumprimida a atual exibição de bloco.
```

Exemplo 4: Comportamento de exibição em diferentes combinações do DISPLON/DISPLOF



① Na atual exibição de bloco são mostradas as linhas de programa de peça do nível de programa 0.
② Na atual exibição de bloco são mostradas as linhas de programa de peça do nível de programa 3.
③ Na atual exibição de bloco são mostradas as linhas de programa de peça do nível de programa 3.
④ Na atual exibição de bloco são mostradas as linhas de programa de peça do nível de programa 7/8.

1.25.2.7 Identificar subrotinas com preparação (PREPRO)

Função

Com a palavra-chave PREPRO todos os arquivos podem ser identificados no fim da linha da instrução PROC durante a inicialização.

Indicação

Este tipo de preparação de programa depende do ajuste do dado de máquina correspondente. Observar as informações do fabricante da máquina.

Literatura:
Manual de funções especiais; Pré-processamento (V2)

Sintaxe

```
PROC … PREPRO
```

Significado

PREPRO:	Palavra-chave para identificação de todos arquivos preparados na inicialização, os programas NC armazenados nos diretórios dos ciclos

Carregar subrotinas com preparação e chamada de subrotina

Tanto as subrotinas preparadas com parâmetros na inicialização como na chamada de subrotina os diretórios dos ciclos são tratados na mesma ordem:

1. _N_CUS_DIR ciclos de usuário
2. _N_CMA_DIR ciclos do fabricante
3. _N_CST_DIR ciclos padrão

No caso de programas NC de mesmo nome e diferente formação, ativa-se a primeira instrução PROC encontrada e a outra instrução PROC será saltada sem mensagem de alarme.

1.25.2.8 Salto de retorno para a subrotina M17

Função

No fim de uma subrotina existe um comando de salto de retorno `M17` (ou o comando de fim de programa de peça `M30`). Ele executa o salto de retorno para o programa de chamada no bloco de programa de peça após a chamada da subrotina.

Indicação

O `M17` e o `M30` são tratados com o mesmo valor na linguagem NC.

Sintaxe

```
PROC <nome de programa>
...
M17/M30
```

Condições gerais

Efeito do salto de retorno para subrotina no modo de controle da trajetória

Se o `M17` (ou o `M30`) estiver sozinho no bloco de programa de peça, então um modo de controle da trajetória que estiver ativo no canal será interrompido.

Para evitar que o modo de controle da trajetória seja interrompido, o `M17` (ou o `M30`) deve ser escrito no último bloco de deslocamento. Adicionalmente, o seguinte dado de máquina deve ser ajustado para "0":

MD20800 $MC_SPF_END_TO_VDI = 0 (sem saída de M30/M17 para a interface NC/PLC)

Exemplo

1. Subrotina com `M17` em um bloco próprio

```
Código de programa              Comentário
N10 G64 F2000 G91 X10 Y10
N20 X10 Z10
N30 M17                         ; Salto de retorno com interrupção do modo de
                                  controle da trajetória.
```

2. Subrotina com `M17` no último bloco de deslocamento

```
Código de programa              Comentário
N10 G64 F2000 G91 X10 Y10
N20 X10 Z10 M17                 ; Salto de retorno sem interrupção do modo de
                                  controle da trajetória.
```

1.25.2.9 Salto de retorno para subrotina RET

Função

Como substituto para o comando de salto de retorno M17 também pode ser utilizado o comando RET. O RET deve ser programado em um bloco de programa de peça próprio. Como o M17 o RET executa o salto de retorno para o programa de chamada no bloco de programa de peça após a chamada da subrotina.

Indicação

Através da programação de parâmetros é possível alterar o comportamento do salto de retorno do RET (veja "Salto de retorno parametrizável da subrotina (RET ...) (Página 175)").

Aplicação

A instrução RET deve ser utilizada quando um modo de controle da trajetória G64 (G641 ... G645) **não** deve ser interrompido através do salto de retorno.

Pré-requisito

O comando RET somente pode ser utilizado em subrotinas que não foram definidas com o atributo SAVE.

Sintaxe

```
PROC <nome de programa>
...
RET
```

Exemplo

Programa principal:

Código de programa	Comentário
PROC MAIN_PROGRAM	; Início do programa
...	
N50 SUB_PROG	; Chamada de subrotina: SUB_PROG
N60 ...	
...	
N100 M30	; Fim do programa

Subrotina:

Código de programa	Comentário
PROC SUB_PROG	
...	
N100 RET	; O salto de retorno é realizado no bloco N60 no programa principal.

1.25.2.10 Salto de retorno parametrizável da subrotina (RET ...)

Função

Geralmente, a partir de uma subrotina com próprio fim de subrotina `RET` ou `M17` se salta retornando ao programa de onde foi realizada a chamada da subrotina, e a usinagem é continuada com a linha após a chamada da subrotina.

Também existem casos de aplicações onde a execução do programa deve continuar em outro ponto, p. ex.:

- Continuação da execução do programa após a chamada do ciclo de desbaste em modo de dialeto ISO (após a descrição do contorno).
- Salto de retorno ao programa principal a partir de qualquer nível da subrotina (também após a ASUP) para tratamento de erros.
- Salto de retorno através de vários níveis de programa para aplicações especiais em ciclos de compilação e em modo de dialeto ISO.

Em tais casos o comando `RET` é programado junto com os "Parâmetros de salto de retorno".

Sintaxe

```
RET("<bloco de destino>")
RET("<bloco de destino>",<bloco após bloco de destino>)
RET("<bloco de destino>",<bloco após bloco de
destino>,<número de níveis de salto de retorno>)
RET("<bloco de destino>", ,<número de níveis de salto de retorno>)
RET("<bloco de destino>",<bloco após bloco de
destino>,<número de níveis de salto de retorno>),
<salto de retorno ao início do programa>)
RET( , ,<número de níveis de salto de retorno>, <salto de
retorno ao início do programa>)
```

Significado

`RET`: Fim de subrotina (utilização ao invés do `M17`)

`<bloco de destino>`: Parâmetro de salto de retorno 1

Nomeia como destino de salto o bloco em que se deve continuar a execução do programa.

Se o parâmetro de salto de retorno 3 não for programado, então o destino do salto se encontra no programa de onde a atual subrotina foi chamada.

As possíveis indicações são:

"<número de bloco>"	Número do bloco de destino
"<marcador de salto>"	Marcador de salto que deve ser colocado no bloco de destino.
"<seqüência de caracteres>"	Seqüência de caracteres que deve ser conhecida no programa (p. ex. nome do programa ou nome da variável). Para a programação da seqüência de caracteres no bloco de destino são aplicadas as seguintes regras: • **Espaço vazio no fim** (em contrapartida ao marcador de salto, que é identificado com um ":" no final). • **Antes da seqüência de caracteres** somente pode ser colocado um número de bloco e/ou um marcador de salto, **nenhum comando de programa**.

`<bloco após bloco de destino>`: Parâmetro de salto de retorno 2

Refere-se ao parâmetro de salto de retorno 1.

Tipo: INT

Valor:		
	0	O salto de retorno é realizado no bloco que foi especificado com o parâmetro de salto de retorno 1.
	> 0	O salto de retorno é realizado no bloco seguinte do bloco que foi especificado com o parâmetro de salto de retorno 1.

`<número de níveis de saltos de retorno>`:

Parâmetro de salto de retorno 3

Nomeia o número de níveis que deverão ser saltados de volta para alcançar o nível de programa em que a execução do programa deverá ser continuada.

Tipo: INT

Valor:
- 1 O programa é continuado "no atual nível do programa - 1" (como `RET` sem parâmetros).
- 2 O programa é continuado no "atual nível do programa - 2", isto é, é realizado o salto de um nível.
- 3 O programa é continuado no "atual nível do programa - 3", isto é, é realizado o salto de dois níveis.
- ...

Faixa de valores: 1 ... 15

`<salto de retorno ao início do programa>`:

Parâmetro de salto de retorno 4

Tipo: BOOL

Valor:
- 1 Se o salto de retorno for realizado ao programa principal e ali estiver ativo um **modo de dialeto ISO**, é realizada a bifurcação para o início do programa.

Indicação

Em um salto de retorno da subrotina com uma seqüência de caracteres, como indicação para a localização de destino sempre se busca por um marcador de salto no programa de chamada.

Se um destino de salto deve ser definido claramente através de uma seqüência de caracteres, a seqüência de caracteres não pode coincidir com o nome de um marcador de salto, pois o salto de retorno da subrotina sempre será executado para o marcador de salto e não para a seqüência de caracteres (veja o exemplo 2).

Condições gerais

Para o salto de retorno através de vários níveis de programa são avaliadas as instruções SAVE dos diversos níveis de programa.

Se em um retorno através de vários níveis de programa uma subrotina modal estiver ativa, e, se em uma das subrotinas saltadas foi programado o comando de desseleção MCALL para a subrotina modal, então a subrotina modal ainda permanece ativa.

⚠ CUIDADO

O programador deve prestar atenção para que no salto de retorno através de vários níveis de programa a continuação seja realizada com os ajustes modais corretos. Isto é obtido, por exemplo, com a programação de um bloco principal correspondente.

Exemplos

Exemplo 1: Continuação no programa principal após a execução da ASUP

```
Programação                                  Comentário
N10010 CALL "UP1"                            ; Nível de programa 0 (programa
                                               principal)
  N11000 PROC UP1                            ; Nível de programa 1
  N11010 CALL "UP2"
    N12000 PROC UP2                          ; Nível de programa 2
    ...
      N19000 PROC ASUP                       ; Nível de programa 3 (execução da
                                               ASUP)
      ...
      N19100 RET("N10900", ,$P_STACK)        ; Salto de retorno da subrotina
N10900                                       ; Continuação no programa principal.
N10910 MCALL                                 ; Desativa a subrotina modal.
N10920 G0 G60 G40 M5                         ; Corrige outros ajustes modais.
```

Exemplo 2: Seqüência de caracteres (<String>) como indicação para a localização do bloco de destino

Programa principal:

```
Código de programa                  Comentário
PROC MAIN_PROGRAM
N1000 DEF INT iVar1=1, iVar2=4
N1010 ...
N1200 subProg1                      ; Chamada da subrotina "subProg1"
N1210 M2 S1000 X10 F1000
N1220 ......
N1400 subProg2                      ; Chamada da subrotina "subProg2"
```

Código de programa	Comentário
N1410 M3 S500 Y20	
N1420 ..	
N1500 lab1: iVar1=R10*44	
N1510 F500 X5	
N1520 ...	
N1550 subprog1: G1 X30	; Aqui o "subProg1" está definido como marcador de salto.
N1560 ...	
N1600 subProg3	Chamada da subrotina "subProg3"
N1610 ...	
N1900 M30	

Subrotina subProg1:

Código de programa	Comentário
PROC subProg1	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("subProg2")	; Salto de retorno ao programa principal no bloco N1400

Subrotina subProg2:

Código de programa	Comentário
PROC subProg2	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("iVar1")	; Salto de retorno ao programa principal no bloco N1500

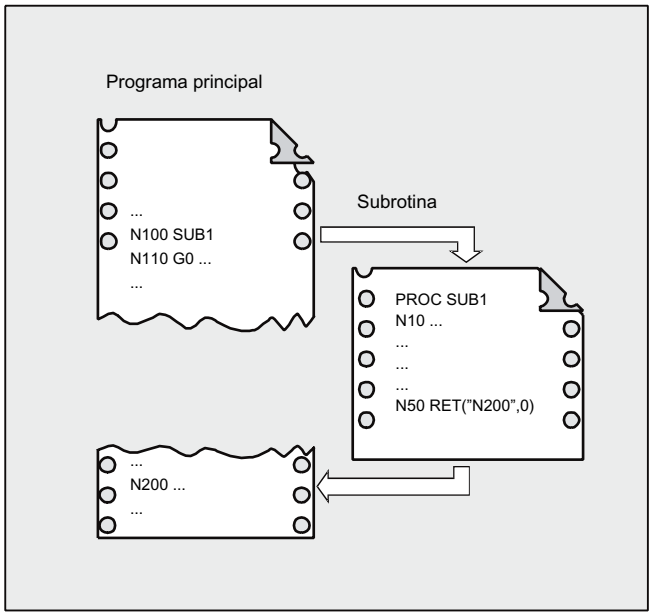
Subrotina subProg3:

Código de programa	Comentário
PROC subProg3	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("subProg1")	; Salto de retorno ao programa principal no bloco N1550

Outras informações

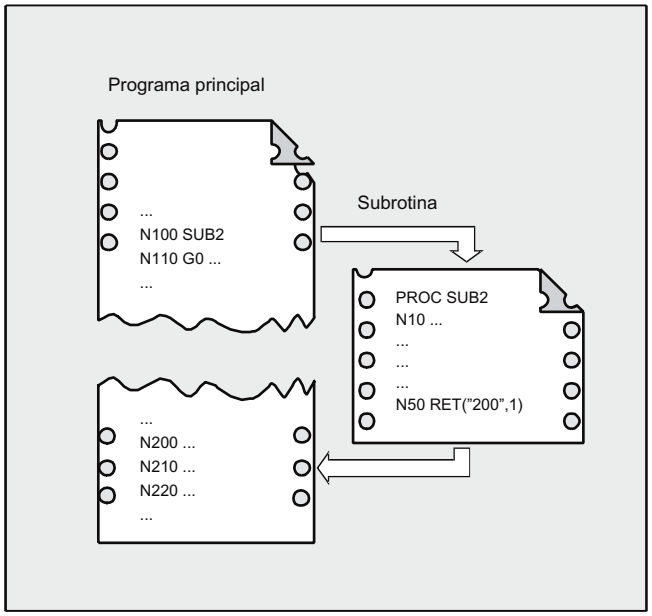
As figuras a seguir devem simplificar a visualização dos diferentes efeitos dos parâmetros de salto de retorno 1 a 3.

1. Parâmetro de salto de retorno 1 = "N200", parâmetro de salto de retorno 2 = 0



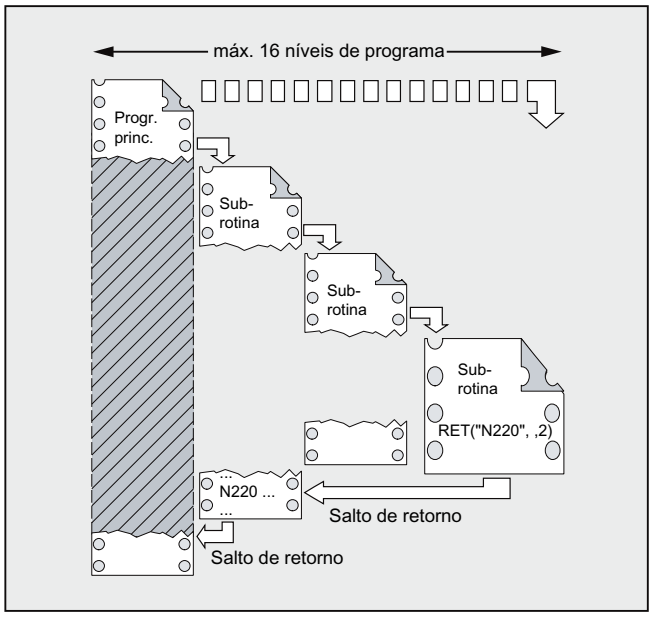
Após o comando `RET` é realizada a continuação da execução do programa com o bloco `N200` no programa principal.

2. Parâmetro de salto de retorno 1 = "N200", parâmetro de salto de retorno 2 = 1



Após o comando `RET` é realizada a continuação da execução do programa com o bloco (`N210`) que segue o bloco `N200` no programa principal.

3. Parâmetro de salto de retorno 1 = "N220", parâmetro de salto de retorno 3 = 2



Após o comando `RET` se realiza o salto retornando dois níveis e a execução do programa é realizada com o bloco `N220`.

1.25.3 Chamada de uma subrotina

1.25.3.1 Chamada de subrotina sem transferência de parâmetros

Função

A chamada de uma subrotina é realizada com o endereço L e o número de subrotina ou através da indicação do nome do programa.

Um programa principal também pode ser chamado como se fosse uma subrotina. Neste caso o fim de programa `M2` ou `M30` definido no programa principal é avaliado como `M17` (fim de programa com retorno para o programa de chamada).

Indicação

Da mesma forma uma subrotina pode ser iniciada como se fosse um programa principal.

Estratégia de localização do comando numérico:

Existe *_MPF?

Existe *_SPF?

Disto resulta: Se o nome da subrotina a ser chamada for idêntico ao nome do programa principal, então o programa principal de chamada é chamado novamente, Este feito, que normalmente não é desejado, deve ser evitado através de escolha clara de nomes através de subrotinas e programas principais.

Indicação

As subrotinas que não requerem nenhuma transferência de parâmetros, também podem ser chamadas a partir de um arquivo de inicialização.

Sintaxe

```
L<número>/<nome de programa>
```

Indicação

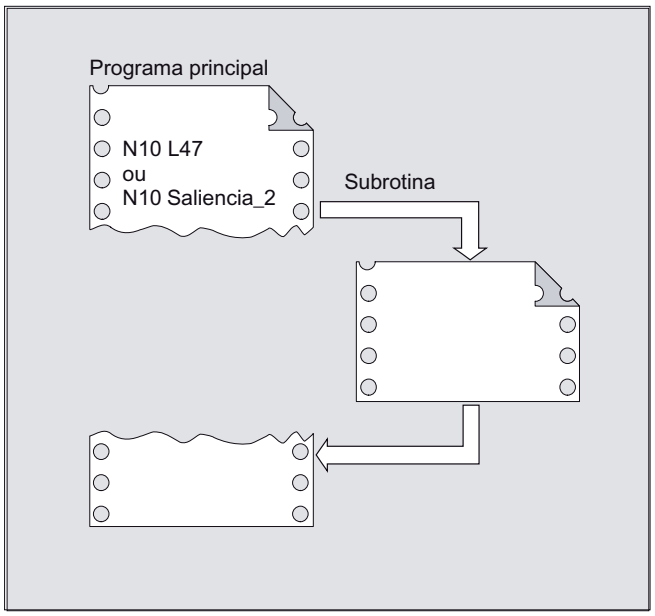
A chamada de uma subrotina sempre deve ser programada em um bloco NC próprio.

Significado

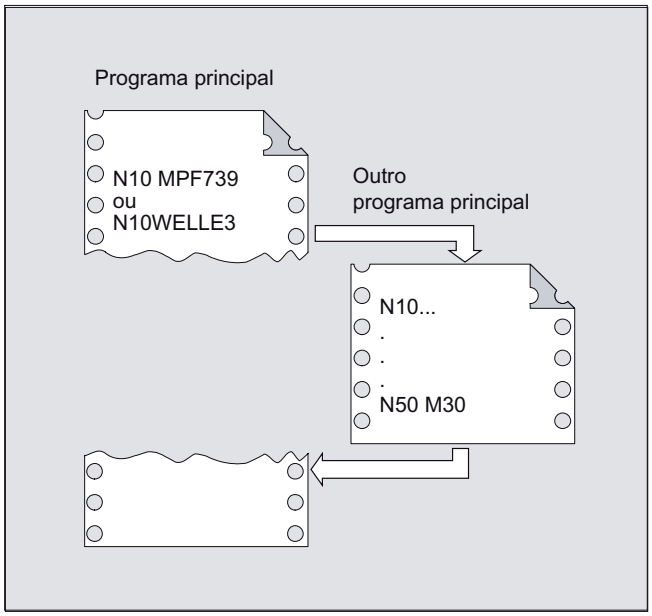
`L`:	Endereço para a chamada da subrotina
`<número>`:	Número da subrotina
	Tipo: INT
	Valor: Máximo de 7 casas decimais
	Atenção: Os zeros à esquerda têm significado na definição de nomes (⇒ L123, L0123 e L00123 são três subrotinas diferentes).
`<nome de programa>`:	Nome da subrotina (ou programa principal)

Exemplos

Exemplo 1: Chamada de uma subrotina sem transferência de parâmetros



Exemplo 2: Chamada de um programa principal como subrotina



1.25.3.2 Chamada de subrotina com transferência de parâmetros (EXTERN)

Função

Na chamada da subrotina com transferência de parâmetros podem ser transferidas variáveis ou valores de forma direta (não para parâmetros VAR).

Subrotinas com transferência de parâmetros devem ser identificadas no programa principal com EXTERN antes da chamada (p. ex. no início do programa). Aqui se especificam o nome da subrotina e os tipos das variáveis na seqüência da transferência.

⚠ CUIDADO
Tanto os tipos de variáveis como a ordem da transferência devem coincidir com as definições declaradas sob PROC na subrotina. Os nomes de parâmetros podem ser diferentes no programa principal e na subrotina.

Sintaxe

```
EXTERN <no do programa>(<Tipo_Par1>,<Tipo_Par2>,<Tipo_Par3>)
...
<nome do programa>(<Valor_Par1>,<Valor_Par2>,<Valor_Par3>)
```

⚠ CUIDADO
A chamada da subrotina sempre deve ser programada em um bloco NC próprio.

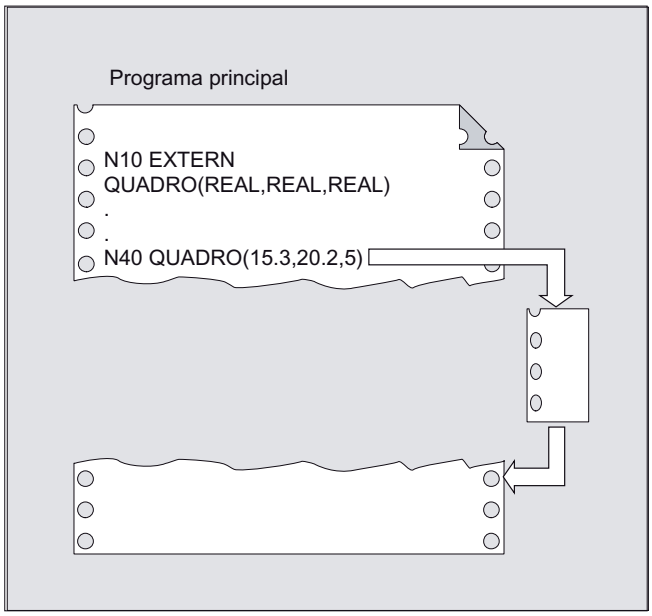
Significado

`<nome de programa>`:	Nome da subrotina
`EXTERN`:	Palavra-chave para a identificação de uma subrotina com transferência de parâmetros **Nota:** O `EXTERN` somente deve ser especificado se a subrotina estiver no diretório de peças ou no diretório de subrotinas global. Os ciclos não precisam ser declarados como `EXTERN`.
`<Tipo_Par1>,<Tipo_Par2>,<Tipo_Par3>`:	Tipos de variáveis dos parâmetros de transferência na seqüência da transferência
`<Valor_Par1>,<Valor_Par2>,<Valor_Par3>`:	Valores de variáveis dos parâmetros de transferência

Exemplos

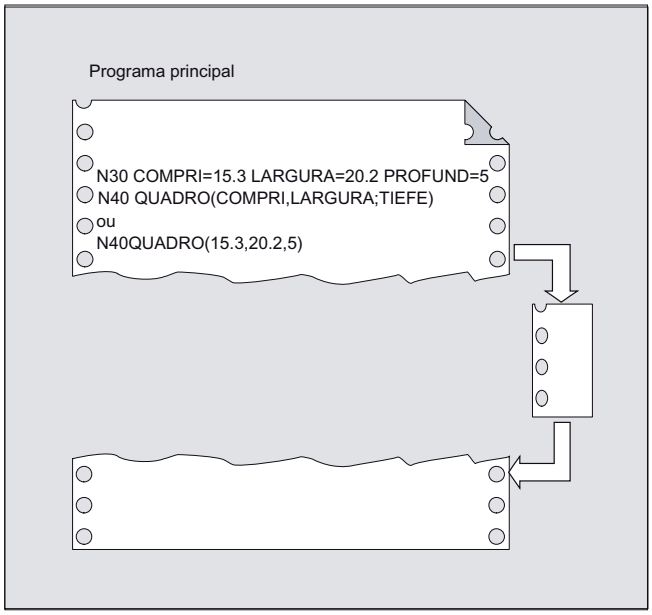
Exemplo 1: Chamada de subrotina com identificação prévia

Código de programa	**Comentário**
`N10 EXTERN QUADRO(REAL,REAL,REAL)`	`; Indicação da subrotina.`
`...`	
`N40 QUADRO(15.3,20.2,5)`	`; Chamada da subrotina com transferência de parâmetros.`



Exemplo 2: Chamada de subrotina sem identificação

Código de programa	Comentário
N10 DEF REAL COMPRIMENTO, LARGURA, PROFUNDIDADE	
N20 …	
N30 COMPRIMENTO=15.3 LARGURA=20.2 PROFUNDIDADE=5	
N40 QUADRO(COMPRIMENTO,LARGURA,PROFUNDIDADE)	; ou: N40 QUADRO(15.3,20.2,5)



1.25.3.3 Número de repetições de programa (P)

Função

Se uma subrotina for executada várias vezes em sucessão, então se pode programar o número desejado de repetições do programa no endereço `P` do bloco com a chamada da subrotina.

> **⚠ CUIDADO**
>
> **Chamada de subrotina com repetição de programa e transferência de parâmetros**
>
> Os parâmetros somente são transferidos na chamada do programa ou primeiro processamento. Para as demais repetições os parâmetros permanecem inalterados. Se houver a necessidade de alterar os parâmetros durante as repetições de programa, então se deve definir os ajustes correspondentes na subrotina.

Sintaxe

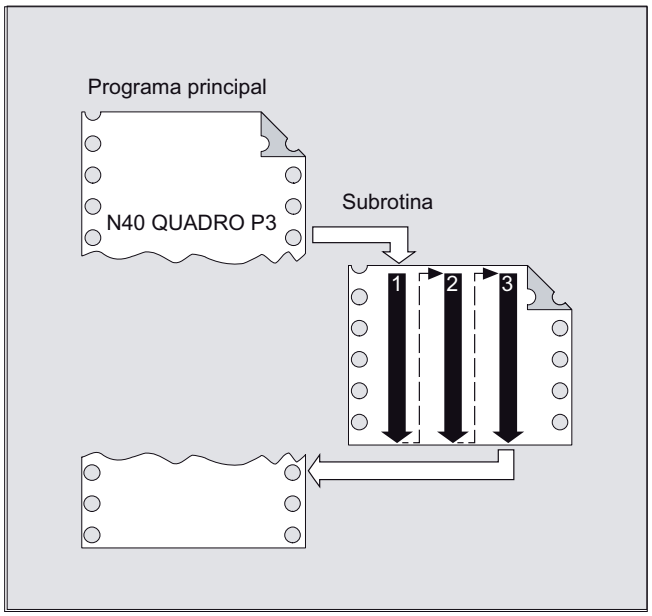
```
<nome de programa> P<valor>
```

Significado

`<nome de programa>`:	Chamada de subrotina	
`P`:	Endereço para programação de repetições de programa	
`<valor>`:	Número de repetições de programa	
	Tipo:	INT
	Faixa de valores:	1 … 9999 (sem sinal)

Exemplo

Código de programa	Comentário
...	
N40 QUADRO P3	; A subrotina QUADRO deve ser executada três vezes consecutivas.
...	



1.25.3.4 Chamada modal de subrotina (MCALL)

Função

Na chamada modal de subrotina com `MCALL` a subrotina é automaticamente chamada e executada após cada bloco com movimento de percurso. Com isso a chamada de subrotinas permite ser automatizada, sendo que ela deve ser executada em diferentes posições da peça (por exemplo para produção de modelos de furação).

A desativação da função é realizada com `MCALL` sem chamada de subrotina ou através da programação de uma nova chamada modal de subrotina para uma nova subrotina.

⚠ CUIDADO

Em uma execução de programa somente uma chamada `MCALL` pode estar ativa ao mesmo tempo. Os parâmetros são transferidos apenas uma vez com a chamada `MCALL`.

Mesmo sem a programação de um movimento, a subrotina modal é chamada nas seguintes situações:

- Na programação dos endereços `S` e `F` quando `G0` ou `G1` estiver ativo.
- Se `G0`/`G1` foi programado sozinho no bloco ou com outros códigos `G`.

Sintaxe

```
MCALL <nome de programa>
```

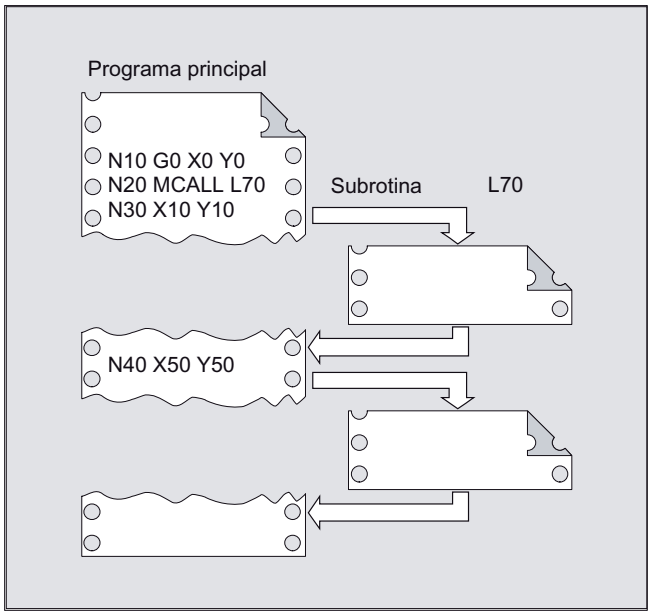
Significado

`MCALL`:	Comando para a chamada de subrotina modal
`<nome de programa>`:	Nome da subrotina

Exemplos

Exemplo 1:

Código de programa	Comentário
N10 G0 X0 Y0	
N20 MCALL L70	; Chamada de subrotina modal.
N30 X10 Y10	; A posição programada é aproximada e depois é executada a subrotina L70.
N40 X50 Y50	; A posição programada é aproximada e depois é executada a subrotina L70.



Exemplo 2:

Código de programa
N10 G0 X0 Y0
N20 MCALL L70
N30 L80

Neste exemplo os blocos NC subseqüentes com eixos de percurso programados estão na subrotina L80. L70 é chamada pela L80.

1.25.3.5 Chamada de subrotinas indireta (CALL)

Função

Em função das condições dadas se pode chamar diferentes subrotinas em uma posição. Para isso se armazena o nome da subrotina em uma variável de tipo STRING. A chamada da subrotina é realizada com `CALL` e o nome da variável.

⚠ CUIDADO

A chamada indireta de subrotinas somente é possível para subrotinas sem transferência de parâmetros. Para a chamada direta de uma subrotina armazene o nome em uma constante de STRING.

Sintaxe

```
CALL <nome do programa>
```

Significado

`CALL`:	Comando para a chamada de subrotina indireta
`<nome de programa>`:	Nome da subrotina (variável ou constante) Tipo: STRING

Exemplo

Chamada direta com constante de STRING:

Código de programa	**Comentário**
…	
`CALL "/_N_WCS_DIR/_N_SUBPROG_WPD/_N_PECA1_SPF"`	`; Chamada direta da subrotina PECA1 com o CALL.`
…	

Chamada indireta através de variável:

```
Código de programa                                              Comentário
…
DEF STRING[100] PROGNAME                                        ; Definição de
                                                                  variável.
PROGNAME="/_N_WCS_DIR/_N_SUBPROG_WPD/_N_PECA1_SPF"              ; Associação da
                                                                  subrotina PECA1 à
                                                                  variável PROGNAME.
CALL PROGNAME                                                   ; Chamada indireta da
                                                                  subrotina PECA1
                                                                  através do CALL e da
                                                                  variável PROGNAME.
…
```

1.25.3.6 Chamada de subrotina indireta com indicação da parte de programa que deve ser executada (CALL BLOCK ... TO ...)

Função

Com `CALL` e a combinação de palavras-chave `BLOCK ... TO` se chama uma subrotina de forma indireta e se executa a parte do programa demarcada entre o marcador inicial e o marcador final.

Sintaxe

```
CALL <nome do programa> BLOCK <marcador inicial> TO <marcador final>
CALL BLOCK <marcador inicial> TO <marcador final>
```

Significado

`CALL`:	Comando para a chamada de subrotina indireta
`<nome de programa>`:	Nome da subrotina (variável ou constante) que contém a parte do programa a ser executada (**Indicação opcional**). Tipo: STRING **Nota:** Se não for programado nenhum `<nome de programa>`, será executada a parte do programa demarcada entre o `<marcador inicial>` e o `<marcador final>` no atual programa.
`BLOCK ... TO ...` :	Combinação de palavras-chave para execução indireta de partes de programa
`<marcador inicial>`:	Variável que faz referência ao início da parte do programa a ser executada. Tipo: STRING
`<marcador final>`:	Variável que faz referência ao fim da parte do programa a ser executada. Tipo: STRING

Exemplo

Programa principal:

Código de programa	Comentário
...	
DEF STRING[20] STARTLABEL, ENDLABEL	; Definição de variáveis para o marcador inicial e marcador final.
STARTLABEL="LABEL_1"	
ENDLABEL="LABEL_2"	
...	
CALL "CONTORNO_1" BLOCK STARTLABEL TO ENDLABEL	; Chamada de subrotina indireta e marcação da parte de programa que deve ser executada.
...	

Subrotina:

Código de programa	Comentário
PROC CONTORNO_1 ...	
LABEL_1	; Marcador inicial: Início da execução da parte do programa
N1000 G1 ...	
...	
LABEL_2	; Marcador final: Fim da execução da parte do programa
...	

1.25.3.7 Chamada indireta de um programa programado em linguagem ISO (ISOCALL)

Função

Com a chamada indireta de programas `ISOCALL` se pode chamar um programa criado em linguagem ISO. Neste caso se ativa o modo ISO ajustado nos dados da máquina. No fim do programa se ativa novamente o modo de execução original. Se nos dados da máquina não foi ajustado nenhum modo ISO, a chamada da subrotina será realizada em modo Siemens.

Para mais informações sobre o modo ISO, veja no(a):
Literatura:
Descrição do funcionamento de dialetos ISO

Sintaxe

```
ISOCALL <nome do programa>
```

Significado

`ISOCALL`:	Palavra-chave para chamada indireta de subrotina que é ativada com o modo ISO ajustado nos dados da máquina
`<nome de programa>`:	Nome de um programa programado em linguagem ISO (variável ou constante de tipo STRING)

Exemplo: Contorno com programação de ciclos chamado a partir do modo ISO

Código de programa	**Comentário**
0122_SPF	; Descrição de contorno em modo ISO
N1010 G1 X10 Z20	
N1020 X30 R5	
N1030 Z50 C10	
N1040 X50	
N1050 M99	
N0010 DEF STRING[5] PROGNAME = "0122"	; Programa de peças Siemens (ciclo)
...	
N2000 R11 = $AA_IW[X]	
N2010 ISOCALL PROGNAME	
N2020 R10 = R10+1	; Executar programa 0122.spf em modo ISO
...	
N2400 M30	

1.25.3.8 Chamada de subrotina com indicação de caminho e parâmetros (PCALL)

Função

Com PCALL se pode chamar subrotinas com indicação absoluta do caminho e transferência de parâmetros.

Sintaxe

```
PCALL <caminho/nome do programa>(<parâmetro 1>,…,<parâmetro n>)
```

Significado

PCALL:	Palavra-chave para chamada de subrotinas com indicação absoluta do caminho.
<caminho/nome de programa>:	Caminho absoluto iniciado com "/" e nome da subrotina incluso. Se não foi especificado nenhum caminho absoluto, o PCALL age como uma chamada padrão de subrotina com identificadores de programa. O identificador de programa é especificado sem prefixo _N_ e sem extensão. Se for necessário programar o nome do programa com prefixo e extensão, então ele deve ser explicitamente declarado com o comando EXTERN com prefixo e extensão.
<parâmetro 1>, ...:	Parâmetro atual de acordo com a instrução PROC da subrotina.

Exemplo

Código de programa

```
PCALL/_N_WCS_DIR/_N_EIXO_WPD/EIXO(parâmetro1,parâmetro2,…)
```

1.25.3.9 Ampliar o caminho de localização na chamada de subrotinas (CALLPATH)

Função

Com o comando CALLPATH se pode ampliar o caminho de localização da chamada de subrotinas.

Dessa forma também podem ser chamadas subrotinas de um diretório de peças de trabalho não selecionado, sem precisar especificar o caminho completo da subrotina.

A ampliação do caminho de localização é realizada antes da entrada para ciclos de usuário (_N_CUS_DIR).

Através dos seguintes eventos se desfaz a seleção do caminho de localização:

- CALLPATH com espaços vazios
- CALLPATH sem parâmetros
- Fim do programa de peça
- Reset

Sintaxe

```
CALLPATH("<nome do caminho>")
```

Significado

CALLPATH:	Palavra-chave para a ampliação programável do caminho de localização. Se programa em uma linha própria do programa de peça.
<nome de caminho>:	Constante ou variável de tipo STRING. Contém a indicação absoluta do caminho de um diretório que deve ampliar o caminho de localização. A indicação do caminho é iniciada com "/". O caminho deve ser especificado completo com prefixo e sufixo. O tamanho máximo do caminho é de 128 Bytes. Se o <nome do caminho> contém um espaço vazio ou se o CALLPATH for chamado sem parâmetros, a instrução do caminho de localização será resetada novamente.

Exemplo

Código de programa

```
CALLPATH("/_N_WCS_DIR/_N_MYWPD_WPD")
```

Com isso se configura o seguinte caminho de localização (a posição 5. é nova):

1. Atual diretório/identificador de subrotina
2. Atual diretório/identificador de subrotina_SPF
3. Atual diretório/identificador de subrotina_MPF
4. /_N_SPF_DIR/identificador de subrotina_SPF
5. **/_N_WCS_DIR/_N_MYWPD/ identificador de subrotina_SPF**
6. /N_CUS_DIR/_N_MYWPD/identificador de subrotina_SPF
7. /_N_CMA_DIR/identificador de subrotina_SPF
8. /_N_CST_DIR/identificador de subrotina_SPF

Condições gerais

- O CALLPATH verifica se os caminhos programados realmente existem. Em caso de erro, a execução do programa de peças será cancelada com o alarme de bloco de correção 14009.
- O CALLPATH também pode ser programado em arquivos INI. Ele atua durante o período de execução do arquivo INI (arquivo INI WPD ou programa de inicialização para dados de NC ativos, p. ex. Frames no 1º canal _N_CH1_UFR_INI). Em seguida o caminho de localização é resetado novamente.

1.25.3.10 Executar subrotina externa (EXTCALL)

Função

Com o comando EXTCALL é possível recarregar e executar uma subrotina a partir de uma memória de programas externa (unidade local, unidade de rede, unidade USB).

O caminho para o diretório externo de subrotinas pode ser pré-configurado com o dado de ajuste:

SD42700 $SC_EXT_PROG_PATH

Junto com o caminho ou identificador de subrotina indicado na chamada EXTCALL se forma o caminho geral do programa chamado.

Indicação

As subrotinas externas não podem conter instruções de salto como GOTOF, GOTOB, CASE, FOR, LOOP, WHILE ou REPEAT.

As construções IF-ELSE-ENDIF são possíveis.

Chamadas de subrotina e chamadas IF-ELSE-ENDIF aninhadas também são permitidas.

Sintaxe

```
EXTCALL("<caminho/><nome do programa>")
```

Significado

`EXTCALL`:	Comando para chamada de uma subrotina externa	
`"<caminho/><nome de programa>"`:	Constante/variável de tipo STRING	
	`<caminho/>`:	Indicação de caminho absoluta ou relativa (**opcional**)

	`<nome de programa>`:	O nome do programa é especificado sem o prefixo "_N_".
		A extensão de arquivo ("MPF", "SPF") pode ser adicionada em nomes de programa com o caractere "_" ou "." (**opcional**).
		Exemplo:
		`"EIXO"`
		ou
		`"EIXO_SPF"` e `"EIXO.SPF"`

Indicação

Indicação de caminho: Abreviações

Na indicação do caminho podem ser utilizadas as seguintes abreviações:

- **LOCAL_DRIVE:** para unidade local
- **CF_CARD:** para cartão CompactFlash
- **USB:** para entrada de USB frontal

CF_CARD: e **LOCAL_DRIVE:** podem ser utilizados como alternativa.

Indicação

Execução externa através da unidade USB

Se os programas devem ser transferidos de uma unidade externa USB através da interface USB, então somente deve ser utilizada a interface de nome "TCU_1" através do X203.

ATENÇÃO

Execução externa através de USB-FlashDrive (na conexão USB frontal)

Não se recomenda uma execução direta a partir de um USB-FlashDrive.

Não existe nenhuma segurança contra problemas de contato, ejeção, quebra por batidas ou retirada por descuido do USB-FlashDrive durante o processamento.

A desconexão durante o trabalho da ferramenta gera uma parada imediata da usinagem e com isso também são provocados danos à peça de trabalho.

Exemplo

Execução a partir de unidade local

Programa principal:

Código de programa

```
N010 PROC MAIN
N020 ...
N030 EXTCALL ("DESBASTE")
N040 ...
N050 M30
```

Subrotina externa:

Código de programa

```
N010 PROC DESBASTE
N020 G1 F1000
N030 X= ... Y= ... Z= ...
N040 ...
...
...
N999999 M17
```

O programa principal "MAIN.MPF" encontra-se na memória NC e é selecionado para execução:

A subrotina "DESBASTE.SPF" que deve ser recarregada, e consequentemente o "DESBASTE.MPF", encontra-se na unidade local no diretório de peças "/user/sinumerik/data/prog/WKS.DIR/WST1.WPD".

O caminho para a subrotina está pré-configurado no SD42700:

SD42700 $SC_EXT_PROG_PATH = "LOCAL_DRIVE:WKS.DIR/WST1.WPD"

Indicação

Sem a indicação do caminho no SD42700 a instrução `EXTCALL` deste exemplo deve ser programada como segue:

```
EXTCALL("LOCAL_DRIVE:WKS.DIR/WST1.WPD/DESBASTE")
```

Outras informações

Chamada EXTCALL com indicação absoluta do caminho

Se a subrotina estiver disponível no caminho especificado, então ela será executada após a chamada `EXTCALL`. E ela não estiver disponível, então a execução do programa será cancelada.

Chamada EXTCALL com indicação relativa do caminho / sem indicação de caminho

Em uma chamada `EXTCALL` com indicação relativa do caminho ou sem a indicação do caminho é realizada uma localização nas memórias disponíveis de programas de acordo com o seguinte modelo:

- Se no SD42700 $SC_EXT_PROG_PATH houver um caminho pré-configurado, então a localização ocorre primeiro neste caminho segundo a indicação na chamada `EXTCALL` (nome do programa e eventualmente um caminho relativo). O caminho absoluto resulta da sequência de caracteres dada pelo(a):
  - caminho pré-configurado no SD42700
  - caractere "/" como caractere de separação
  - o caminho ou identificador de subrotina especificado com `EXTCALL`
- Se a subrotina chamada não foi encontrada no caminho pré-configurado, será executada a localização nos diretórios da memória de usuário conforme a indicação na chamada `EXTCALL`.
- A localização é encerrada assim que a subrotina for encontrada pela primeira vez. Se a localização não encontrar nenhum resultado, então o programa será cancelado.

Memória temporária configurável (memória FIFO)

Para a execução de um programa no modo "Executar externamente" (programa principal ou subrotina) é necessária uma memória temporária no NCK. O tamanho da memória de recarregamento é pré-configurada com 30 kByte e somente pode ser alterada pelo fabricante da máquina, como outros dados de máquina relevantes à memória.

Para os programas (principais ou subrotinas) que são processados simultaneamente no modo "Executar externamente", deve-se configurar uma memória de recarregamento para cada um.

RESET, POWER ON

Com RESET e POWER ON as chamadas externas de subrotinas são canceladas e a memória temporária é apagada.

Uma subrotina selecionada para modo "Executar externamente" permanece selecionada além do RESET / fim do programa para o modo "Executar externamente". A seleção se perde com um POWER ON.

Literatura

Para mais informações sobre "Execução externa", veja:

Manual de funções básicas; BAG, Canal, Operação do programa, Comportamento Reset (K1)

1.25.4 Ciclos

1.25.4.1 Ciclos: Parametrizar ciclos de usuário

Função

Com os arquivos cov.com e uc.com podem ser parametrizados ciclos próprios.

O arquivo cov.com é fornecido com os ciclos padronizados e pode ser ampliado de acordo. O arquivo uc.com deve ser criado pelo próprio usuário.

Ambos arquivos devem ser armazenados no sistema passivo de arquivos no diretório "Ciclos de usuário" (ou com a indicação correspondente do caminho):

```
;$PATH=/_N_CUS_DIR
```

no programa.

Arquivos e caminhos

`cov.com_COM`	Visão geral dos ciclos
`uc.com`	Descrição da chamada de ciclo

Adaptação do cov.com - Visão geral dos ciclos

O arquivo cov.com fornecido junto com os ciclos padronizados possui a seguinte estrutura:

`%_N_COV_COM`	Nome de arquivo
`;$PATH=/_N_CST_DIR`	Indicação de caminho
`;Vxxx 11.12.95 Sca Vista geral de ciclos`	Linha de comentário
`C1(CYCLE81) furação, centragem`	Chamada do 1º ciclo
`C2(CYCLE82) furação, escareamento plano`	Chamada do 2º ciclo
`...`	
`C24(CYCLE98) seqüências de roscas`	Chamada do último ciclo
`M17`	Fim do arquivo

Sintaxe

Para cada novo ciclo adicionado se deve adicionar uma linha com a seguinte sintaxe:

```
C<número> (<nome do ciclo>) texto do comentário
```

Número: um número inteiro qualquer que ainda não tenha sido utilizado no arquivo;

Nome do ciclo: o nome do programa do ciclo a ser incluso

Texto do comentário: um texto de comentário opcional para o ciclo

Exemplo:

```
C25 (MEU_CICLO_1) ciclo de usuário_1
C26 (CICLO ESPECIAL)
```

Exemplo de arquivo uc.com - Descrição dos ciclos de usuário

A explicação está baseada na continuação do exemplo:
Para os seguintes dois ciclos se deve criar uma nova parametrização de ciclo:

```
Programação            Comentário
PROC MEU_CICLO_1 (REAL PAR1, INT PAR2, CHAR PAR3, STRING[10] PAR4)
O ciclo possui os seguintes parâmetros de transferência:
PAR1:                  ; valor real na faixa de -1000.001 <= PAR2 <= 123.456, ocupação
                         prévia com 100
PAR2:                  ; valor inteiro positivo entre 0 <= PAR3 <= 999999, ocupação
                         prévia com 0
PAR3:                  ; 1 caractere ASCII
PAR4:                  ; String de tamanho 10 para um nome de subrotina
...
M17                    ;
```

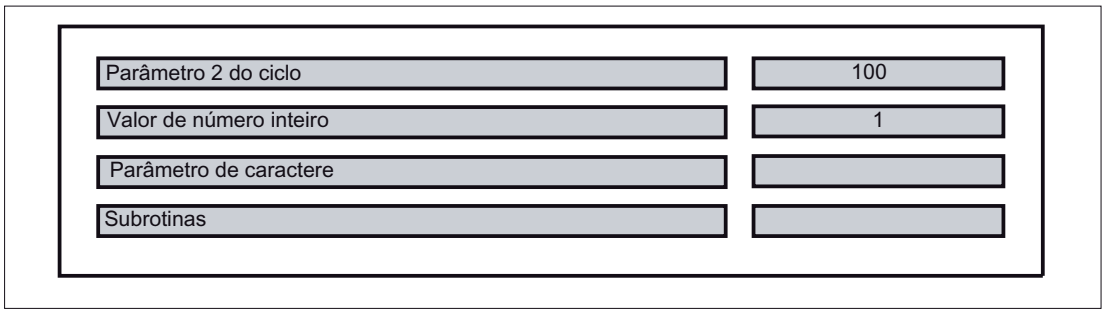
```
Programação            Comentário
PROC CICLOESPECIAL (REAL VALOR1, INT VALOR2)
O ciclo possui os seguintes parâmetros de transferência:
VALOR1:                ; valor Real sem restrição de faixa de valores e ocupação
                         prévia
VALOR2:                ; valor inteiro sem restrição de faixa de valores e ocupação
                         prévia
...
M17                    ;
```

Arquivo uc.com associado:

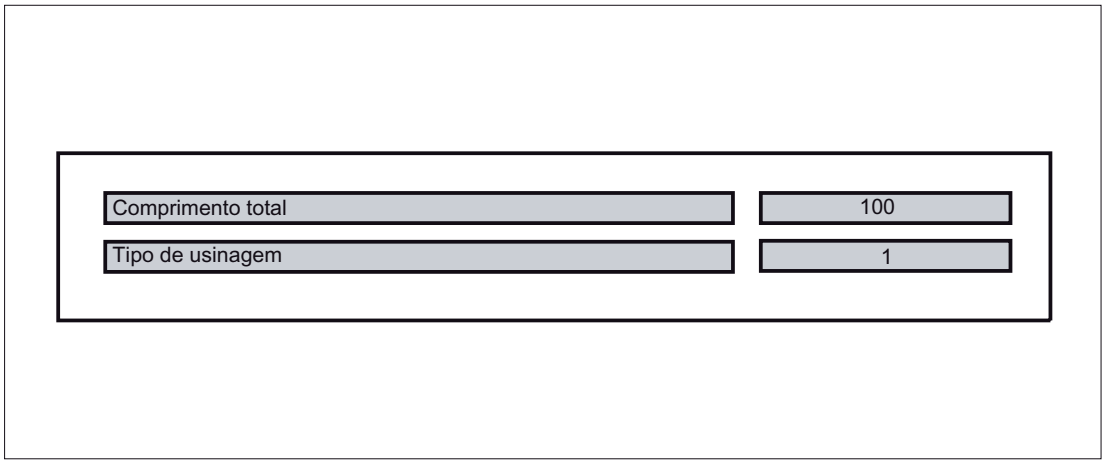
```
Programação
%_N_UC_COM
;$PATH=/_N_CUS_DIR
//C25(MEU_CICLO_1) ciclo de usuário_1
(R/-1000.001 123.456 / 100 /Parâmetro_2 do ciclo)
(I/0 999999 / 1 / valor inteiro)
(C//"A" / parâmetro de caractere)
(S///nome da subrotina)
//C26(CICLO ESPECIAL)
(R///tamanho total)
(I/*123456/3/tipo de processamento)
M17
```

Exemplo dos dois ciclos

Tela de exibição do ciclo `MEU_CICLO_1`



Tela de exibição do ciclo `CICLOESPECIAL`



Descrição da sintaxe para o arquivo uc.com - Descrição dos ciclos de usuário

Linha de cabeçalho por ciclo:

igual como no arquivo cov.com e precedido de "//"

```
//C <número> (<nome do ciclo>) texto do comentário
```

Exemplo:

```
//C25 (MEU_CICLO_1) ciclo do usuário_
```

Linha de descrição por parâmetro:

```
(<tipo de arquivo> / <valor mínimo> <valor máximo>
/ <valor de ocupação prévia> / <comentário>)
```

Identificador do tipo de arquivo:

R	para Real
I	para Integer
C	para caracteres (1 caractere)
S	para String

Valor mínimo, valor máximo (pode ser omitido)

Limites do valor especificado que são verificados durante a especificação; os valores fora desta faixa não podem ser especificados. Podem ser especificados valores de enumeração que podem ser operados com a tecla Toggle; estes são enumerados começando com "*", demais valores não permitidos.

Exemplo:

```
(I/*123456/1/tipo de processamento)
```

Nos tipos String e caractere não existem limites.

Valor de ocupação prévia (pode ser omitido)

Valor pré-definido na tela correspondente na chamada do ciclo; ele pode ser alterado pelo operador.

Comentário

Texto com no máximo 50 caracteres que é exibido na tela de chamada do ciclo antes do campo de entrada do parâmetro.

1.26 Técnica de macros (DEFINE ... AS)

⚠ CUIDADO
Com a técnica de macros se pode alterar consideravelmente a linguagem de programação do comando! Por isso aplique esta técnica de macros com extremo cuidado!

Função

Como macro definimos o agrupamento de instruções individuais como uma instrução geral de nome próprio. As funções G, M e H ou as subrotinas L também podem ser aplicadas como macros. Na chamada da macro na execução do programa os comandos programados no nome da macro são executados seqüencialmente.

Aplicação

Seqüências de instruções que se repetem, são programadas apenas uma vez em forma de macro em um módulo de macro (arquivo de macro) separado ou uma vez no início do programa. A macro pode ser chamada e executada em qualquer programa principal ou subrotina.

Ativação

Para utilizar as macros de um arquivo de macro no programa NC se deve carregar o arquivo de macro no NC.

Sintaxe

Definição de macro:
```
DEFINE <nome de macro> AS <instrução 1> <instrução 2> ...
```
Chamada no programa NC:
```
<nome de macro>
```

Significado

`DEFINE ... AS` :	Combinação de palavras-chave para definição de uma macro
`<nome de macro>`:	Nome da macro
	Como nomes de macros somente são permitidos identificadores.
	Com o nome da macro se chama a macro a partir do programa NC.
`<instrução>`:	Instrução de programação que deve estar contida na macro.

Regras para definição de macros

- Na macro se pode definir qualquer identificador, funções G, M e H e nomes de programas L.
- As macros também podem ser definidas no programa NC.
- As macros de função G somente podem ser definidas globalmente no comando em módulos de macro.
- As funções H e L são programáveis com 2 dígitos.
- As funções M e G podem ser programadas com 3 dígitos.

⚠ CUIDADO
Palavras-chave e nomes reservados não podem ser redefinidos com macros.

Condições gerais

Não é possível fazer um aninhamento de macros.

Exemplos

Exemplo 1: Definição de macro no início do programa

Código de programa	Comentário
`DEFINE LINHA AS G1 G94 F300`	`; Definição de macro`
`...`	
`...`	
`N70 LINHA X10 Y20`	`; Chamada de macro`
`...`	

Exemplo 2: Definições de macro em um arquivo de macro

Código de programa	Comentário
`DEFINE M6 AS L6`	`; Para a troca de ferramentas se chama uma subrotina que assume a transferência de dados necessária. Na subrotina é executada a função M de troca de ferramentas propriamente dita (p. ex. M106).`
`DEFINE G81 AS DRILL(81)`	`; Simulação da função G DIN.`
`DEFINE G33 AS M333 G333`	`; Para a usinagem de roscas se solicita a sincronização com o PLC. A função G33 original foi renomeada para G333 pelo MD, a programação permanece a mesma para o usuário.`

Exemplo 3: Arquivo de macro externo

Após o carregamento do arquivo de macro externo no comando, o arquivo de macro deve ser carregado no NC. Somente então se pode utilizar as macros no programa NC.

```
Código de programa                Comentário
%_N_UMAC_DEF
;$PATH=/_N_DEF_DIR               ; Macros específicas de usuário
DEFINE PI AS 3.14
DEFINE TC1 AS M3 S1000
DEFINE M13 AS M3 M7              ; Fuso gira à direita, líquido refrigerante ligado
DEFINE M14 AS M4 M7              ; Fuso gira à esquerda, líquido refrigerante ligado
DEFINE M15 AS M5 M9              ; Fuso parado, líquido refrigerante desligado
DEFINE M6 AS L6                  ; Chamada do programa de troca de ferramentas
DEFINE G80 AS MCALL              ; Desseleção do ciclo de furação
M30
```

Gerenciamento de arquivos e programas 2

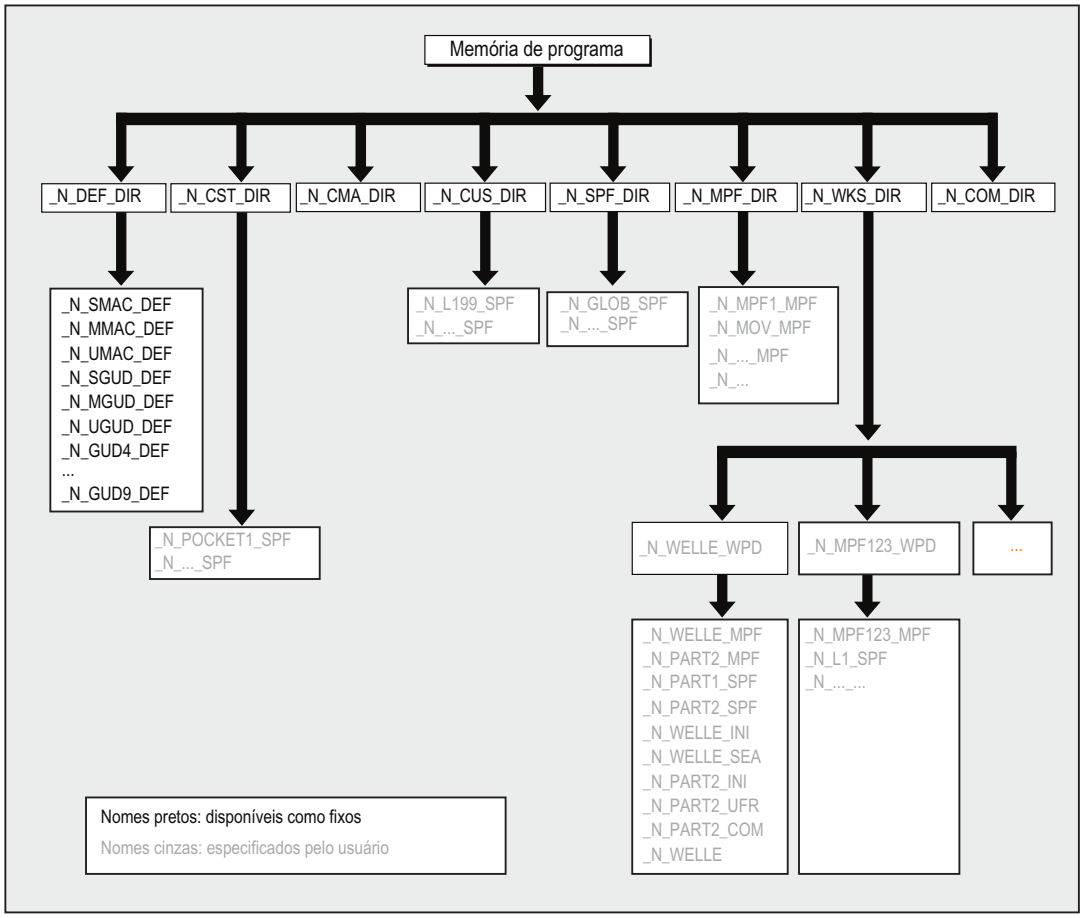
2.1 Memória de programa

Função

Na memória de programas são armazenados arquivos e programas (p. ex. programas principais e subrotinas, definições de macro) de forma permanente (→ Sistema passivo de arquivos).

Literatura:
Manual de funções ampliadas; Configuração de memória (S7)

Também existe uma quantidade de tipos de arquivos que pode ser armazenada temporariamente e, em caso de necessidade (p. ex. para a usinagem de uma determinada peça), estes arquivos são transferidos para a memória de trabalho (p. ex. para inicializações).



Diretórios padrão

Como padrão estão disponíveis os seguintes diretórios:

Diretório	Conteúdo
_N_DEF_DIR	Módulos de dados e módulos de macros
_N_CST_DIR	Ciclos padronizados
_N_CMA_DIR	Ciclos de fabricante
_N_CUS_DIR	Ciclos de usuário
_N_WCS_DIR	Peças de trabalho
_N_SPF_DIR	Subrotinas globais
_N_MPF_DIR	Programas principais
_N_COM_DIR	Comentários

Tipos de arquivo

Na memória de programa podem ser inseridos os seguintes tipos de arquivos:

Tipo de arquivo	Descrição
nome_MPF	Programa principal
nome_SPF	Subrotina
nome_TEA	Dados de máquina
nome_SEA	Dados de ajuste
nome_TOA	Correções de ferramentas
nome_UFR	Deslocamentos de ponto zero/Frame
nome_INI	Arquivo de inicialização
nome_GUD	Dados de usuário globais
nome_RPA	Parâmetros R
nome_COM	Comentário
nome_DEF	Definições de dados de usuário globais e macros

Diretório principal das peças de trabalho (_N_WCS_DIR)

O diretório principal das peças de trabalho está configurado com a denominação `_N_WCS_DIR` na memória de programa. O diretório principal das peças de trabalho contém todos os diretórios correspondentes de todas peças de trabalho que foram programadas.

Diretórios de peça de trabalho (..._WPD)

Para uma manipulação flexível dos dados e programas, determinados dados e programas podem ser agrupados ou armazenados em diferentes diretórios de peça de trabalho.

Um diretório de peça de trabalho contém todos arquivos necessários para a usinagem de uma peça de trabalho. Estes podem ser programas principais, subrotinas, quaisquer programas de inicialização e arquivos de comentários.

Após a seleção do programa, os programas de inicialização são executados uma vez no primeiro Start do programa de peça (de acordo com o dado de máquina MD11280 $MN_WPD_INI_MODE).

Exemplo:

O diretório de peça de trabalho _N_EIXO_WPD, que foi criado para a peça EIXO, contém os seguintes arquivos:

Arquivo	Descrição
_N_EIXO_MPF	Programa principal
_N_PART2_MPF	Programa principal
_N_PART1_SPF	Subrotina
_N_PART2_SPF	Subrotina
_N_EIXO_INI	Programa geral de inicialização dos dados para a peça de trabalho
_N_EIXO_SEA	Programa de inicialização dos dados de ajuste
_N_PART2_INI	Programa geral de inicialização dos dados para o programa Part 2
_N_PART2_UFR	Programa de inicialização dos dados de Frame para o programa Part 2
_N_WELLE_COM	Arquivo de comentários

Criar diretórios de peça de trabalho em um PC externo

O procedimento descrito a seguir é executado em uma estação externa de dados. Para o gerenciamento de arquivos e programas (do PC para o comando) realizado diretamente no comando existem informações contidas em seu manual de operação.

Criar diretórios de peça de trabalho com indicação de caminho ($PATH=…)

Na segunda linha de um arquivo o caminho de destino é indicado com `$PATH=…` O arquivo então é armazenado no caminho especificado.

Exemplo:

Código de programa

```
%_N_EIXO_MPF
;$PATH=/_N_WCS_DIR/_N_EIXO_WPD
N10 G0 X… Z…
...
M2
```

O arquivo _N_EIXO_MPF é armazenado no diretório /_N_WCS_DIR/_N_EIXO_WPD.

Criar diretório de peça de trabalho sem indicação de caminho

Se faltar a indicação do caminho, então os arquivos de extensão _SPF são armazenados no diretório /_NSPF_DIR, arquivos de extensão _INI na memória de trabalho e todos demais arquivos no diretório /_NMPF_DIR.

Exemplo:

Código de programa

```
%_N_EIXO_SPF
...
M17
```

O arquivo _N_EIXO_SPF é armazenado no diretório /_N_SPF_DIR.

Selecionar peça de trabalho para usinagem

Um diretório de peça de trabalho pode ser selecionado para a execução em um canal. Se neste diretório encontra-se um programa principal de **mesmo nome** ou apenas um único programa principal (MPF), então este será automaticamente selecionado para execução.

Literatura:
/BAD/ Manual de operação HMI Advanced; Capítulo "Jobliste" assim como "Selecionar programa para execução"

Caminhos de localização na chamada de subrotinas

Se o caminho de chamada não for especificado explicitamente no programa de peça para chamada de uma subrotina (ou também arquivo de inicialização), então o programa chamado será procurado em um caminho fixo.

Chamada de subrotina com indicação absoluta de caminho

Exemplo:

Código de programa

```
...
CALL"/_N_CST_DIR/_N_CYCLE1_SPF"
...
```

Chamada de subrotina sem indicação absoluta de caminho

Normalmente os programas são chamados sem indicação do caminho.

Exemplo:

Código de programa

```
...
CYCLE1
...
```

Os diretórios são pesquisados para localização do programa chamado na seguinte seqüência:

Nº	Diretório	Descrição
1	atual diretório / *nome*	Diretório principal de peças de trabalho ou diretório padrão _N_MPF_DIR
2	atual diretório / *nome_SPF*	
3	atual diretório / *nome_MPF*	
4	/_N_SPF_DIR / *nome_SPF*	Subrotinas globais
5	/_N_CUS_DIR / *nome_SPF*	Ciclos do usuário
6	/_N_CMA_DIR / *nome_SPF*	Ciclos do fabricante
7	/_N_CST_DIR / *nome_SPF*	Ciclos padronizados

Programar caminhos de localização na chamada da subrotina (CALLPATH)

O caminho de localização para chamada de subrotinas pode ser ampliado com o comando de programa de peças `CALLPATH`.

Exemplo:

```
Código de programa
CALLPATH("/_N_WCS_DIR/_N_MYWPD_WPD")
...
```

O caminho de localização é armazenado antes da posição 5 (ciclo do usuário) conforme programação.

Para mais informações sobre caminhos de localização programáveis na chamada de subrotinas com `CALLPATH`, veja o capítulo "Ampliar o caminho de localização em chamadas de subrotina com CALLPATH".

2.2 Memória de trabalho (CHANDATA, COMPLETE, INITIAL)

Função

A memória de trabalho contém os atuais dados de sistema do usuário, com os quais se opera o comando (sistema de arquivos ativo), p. ex.:

- Dados de máquina ativos
- Dados de corretores de ferramentas
- Deslocamentos de ponto zero
- ...

Programas de inicialização

Aqui trata-se de programas com os quais os dados da memória de trabalho são inicializados. Para isso podem ser utilizados os seguintes tipos de arquivos:

Tipo de arquivo	Descrição
nome_TEA	Dados de máquina
nome_SEA	Dados de ajuste
nome_TOA	Correções de ferramentas
nome_UFR	Deslocamentos de ponto zero/Frame
nome_INI	Arquivo de inicialização
nome_GUD	Dados de usuário globais
nome_RPA	Parâmetros R

Informações sobre todos tipos de dados estão disponíveis no manual de operação da interface de operação.

Áreas de dados

Os dados podem ser organizados em diferentes áreas onde serão aplicados. Por exemplo, um comando pode dispor de vários canais ou, como de costume, dispor de vários eixos.

Existe:

Identificação	Áreas de dados
NCK	Dados específicos do NCK
CH<n>	Dados específicos de canal (<n> indica o número do canal)
AX<n>	Dados específicos de eixo (<n> especifica o número do eixo da máquina)
TO	Dados de ferramenta
COMPLETE	Todos dados

Criar programa de inicialização em um PC externo

Com o auxílio do identificador da área de dados e do tipo de arquivo podem ser determinadas áreas que serão consideradas como uma unidade durante a gravação de segurança dos dados:

_N_AX5_TEA_INI	Dados da máquina para 5 eixos
_N_CH2_UFR_INI	Frames do canal 2
_N_COMPLETE_TEA_INI	Todos dados de máquina

Após a colocação em funcionamento do comando existe um bloco de dados na memória de trabalho que assegura o funcionamento correto do comando.

Procedimento com comandos de vários canais (CHANDATA)

O `CHANDATA(<número do canal>)` para vários canais somente é permitido no arquivo _N_INITIAL_INI. Este é o arquivo de colocação em funcionamento com o qual são inicializados todos dados do comando.

```
Código de programa                          Comentário
%_N_INITIAL_INI
CHANDATA(1)
                                            ; Atribuição de eixo de máquina do canal 1:
$MC_AXCONF_MACHAX_USED[0]=1
$MC_AXCONF_MACHAX_USED[1]=2
$MC_AXCONF_MACHAX_USED[2]=3
CHANDATA(2)
                                            ; Atribuição de eixo de máquina do canal 2:
$MC_AXCONF_MACHAX_USED[0]=4
$MC_AXCONF_MACHAX_USED[1]=5
CHANDATA(1)
                                            ; Dados de máquina de eixo:
                                            ; Janela de parada exata aproximada:
$MA_STOP_LIMIT_COARSE[AX1]=0.2              ; Eixo 1
$MA_STOP_LIMIT_COARSE[AX2]=0.2              ; Eixo 2
                                            ; Janela de parada exata fina:
$MA_STOP_LIMIT_FINE[AX1]=0.01               ; Eixo 1
$MA_STOP_LIMIT_FINE[AX1]=0.01               ; Eixo 2
```

⚠ CUIDADO

Instrução CHANDATA

No programa de peça a instrução `CHANDATA` somente pode ser definida para o canal onde o programa NC será executado; isto é, o comando pode ser utilizado para proteger programas NC para que não sejam executados acidentalmente em outro canal.

A execução do programa será cancelada em caso de erro.

Indicação

Os arquivos INI em listas de tarefas não contém nenhuma instrução `CHANDATA`.

Salvar programas de inicialização (COMPLETE, INITIAL)

Os arquivos da memória de trabalho podem ser salvos em um PC externo e deste recarregados.

- Os arquivos são salvos com `COMPLETE`.
- Com `INITIAL` se gera um arquivo INI (_N_INITIAL_INI) que se estende em todas as áreas.

Carregar programas de inicialização

ATENÇÃO

Quando se carrega o arquivo de nome "INITIAL_INI", todos dados que não são alimentados com o arquivo serão inicializados com dados padrão. Disso estão excluídos apenas os dados de máquina. Os **dados de ajuste, dados de ferramenta, NPV, valores GUD, ...** também são alimentados com dados padrão (normalmente é o "ZERO").

Por exemplo, o arquivo COMPLETE_TEA_INI é apropriado para o carregamento de dados individuais de máquina. Deste arquivo o comando apenas espera os dados de máquina. Dessa forma as demais áreas de dados permanecem inalteradas neste caso.

Carregar programas de inicialização

Os programas INI também podem ser selecionados e executados como programas de peças, isto se utilizarem apenas os dados de um canal. Assim também é possível inicializar dados controlados pelo programa.

2.3 Instrução de estruturação no editor Step (SEFORM)

Função

A instrução de estruturação SEFORM é avaliada no editor Step (suporte de programação baseado em editor), para então gerar a vista do passo para HMI Advanced. A vista do passo serve para facilitar a leitura da subrotina NC.

Sintaxe

```
SEFORM(<nome de secção>,<nível>,<Icon>)
```

Significado

SEFORM()	Chamada de função da instrução de estruturação com os parâmetros <nome de secção>, <nível> e <Icon>
<nome de secção>	Identificador do passo de trabalho Tipo: STRING
<nível>	Índice para o nível principal ou subnível Tipo: INT Valor: 0 Nível principal 1, ..., <n> Subnível 1, ... , Subnível <n>
<Icon>	Nome do ícone que deve ser exibido nesta secção. Tipo: STRING

Indicação

As instruções SEFORM são criadas no editor Step.

A seqüência de caracteres transferida com o parâmetro <nome de secção> é armazenada de forma análoga à instrução MSG, de forma sincronizada com o processamento principal, na variável BTSS. A informação permanece até a próxima instrução SEFORM ser sobrescrita. O conteúdo é apagado com Reset e o fim do programa de peças.

Os parâmetros <nível> e <Icon> são controlados durante a execução do programa de peça, mas não são processados.

Literatura

Para mais informações sobre o suporte de programação baseado em editor, veja no(a): Manual de operação HMI Advanced

Áreas de proteção 3

3.1 Definição das áreas de proteção (CPROTDEF, NPROTDEF)

Função

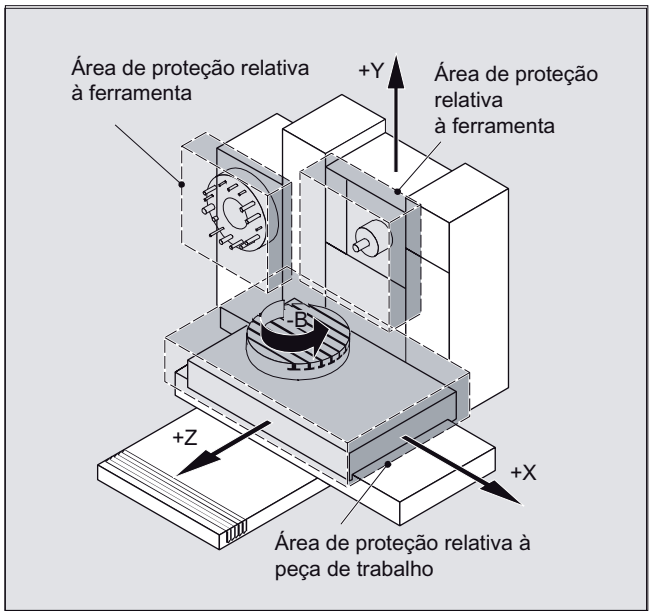
Com a ajuda das áreas de proteção se pode proteger vários elementos da máquina, o equipamento assim como a peça de trabalho de eventuais movimentos incorretos.

Áreas de proteção **relativas à ferramenta**:

Para componentes que pertencem à ferramenta (p. ex. ferramenta, porta-ferramenta).

Áreas de proteção **relativas à peça de trabalho**:

Para componentes que pertencem à peça de trabalho (p. ex. componentes da peça de trabalho, mesa de fixação, garras de fixação, mandril do fuso, contraponta).



Sintaxe

```
DEF INT NOT_USED
G17/G18/G19
CPROTDEF/NPROTDEF(<n>,<t>,<applim>,<applus>,<appminus>)
G0/G1/... X/Y/Z...
...
EXECUTE (NOT_USED)
```

Significado

DEF INT NOT_USED:	Variável local, definir como tipo de dado INTEGER (consulte o cap. "Ações síncronas de movimentos (Página 563)")
G17/G18/G19:	O plano desejado é selecionado antes de CPROTDEF ou NPROTDEF com G17/G18/G19 e não pode ser modificado antes de EXECUTE. Não é permitida uma programação da terceira coordenada entre CPROTDEF ou NPROTDEF e EXECUTE.
CPROTDEF:	Definir áreas de proteção específicas de canal (só para NCU 572/573)
NPROTDEF:	Definir áreas de proteção específicas da máquina
G0/G1/... X/Y/Z... ...:	O contorno das áreas de proteção é especificado com no máximo 11 movimentos de percurso dentro do plano selecionado. Neste caso o primeiro movimento de percurso é o movimento no contorno. Aqui como área de proteção vale a área à esquerda do contorno. **Nota:** Os movimentos de percurso entre CPROTDEF ou NPROTDEF e EXECUTE não são executados, pois eles definem a área de proteção.
EXECUTE:	Finalização da definição
<n>:	Número da área de proteção definida
<t>:	Tipo da área de proteção TRUE: Área de proteção em função da **ferramenta** FALSE: Área de proteção em função da **peça de trabalho**
<applim>:	Tipo de limitação na 3ª dimensão 0: Sem limitação 1: Limitação no sentido positivo 2: Limitação no sentido negativo 3: Limitação nos sentidos positivo e negativo
<applus>:	Valor da limitação no sentido positivo da 3ª dimensão
<appminus>:	Valor da limitação no sentido negativo da 3ª dimensão
NOT_USED:	A variável de erro não tem efeito nas áreas de proteção com EXECUTE

Condições gerais

Durante a definição das áreas de proteção:

- nenhuma correção de raio de fresa ou de raio de corte pode estar ativa,
- nenhuma transformação pode estar ativa.
- nenhum Frame pode estar ativo.

Também não podem ser programadas a aproximação do ponto de referência (G74), aproximação do ponto fixo (G75), parada de localização de blocos ou o fim de programa.

Outras informações

Definição de áreas de proteção

Para definição de áreas de proteção existem:

- `CPROTDEF` para áreas de proteção específicas de canal
- `NPROTDEF` para áreas de proteção específicas da máquina
- Descrição do contorno da área de proteção
- Encerramento da definição com `EXECUTE`

Pode-se deslocar o ponto de referência da área de proteção de modo relativo quando a área de proteção está ativada no programa de peça NC.

Ponto de referência da descrição do contorno

As áreas de proteção relativas à peça de trabalho são definidas no sistema de coordenadas básico.

As áreas de proteção relativas à ferramenta são especificadas em função do ponto de referência do porta-ferramentas F.

Elementos de contorno permitidos

Para a descrição do contorno da área de proteção são permitidos:

- `G0`, `G1` para elementos de contorno retos
- `G2` para segmentos circulares no sentido horário (apenas para área de proteção relativa à peça de trabalho)
- `G3` para segmentos circulares no sentido anti-horário

Indicação

Se for descrita uma circunferência inteira para a área de proteção, então ela deve ser dividida em duas semicircunferências. A ordem `G2`, `G3` ou `G3`, `G2` não é permitida. Aqui deve ser inserido um curto bloco `G1`.

O último ponto da descrição do contorno deve coincidir com o primeiro ponto.

Áreas de proteção externas

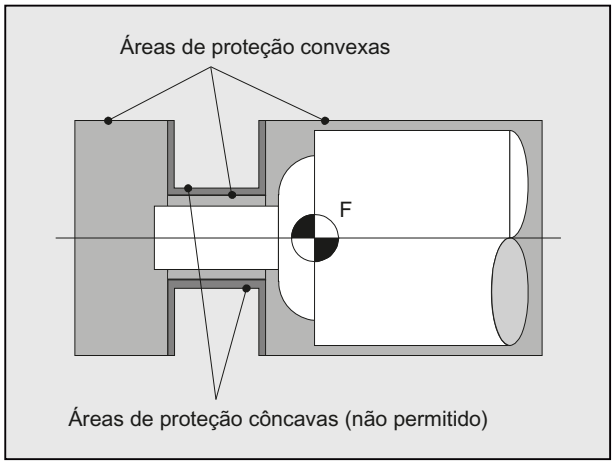
As áreas de proteção externas (possível apenas em áreas de proteção relativas à peça de trabalho) devem ser definidas no sentido horário.

Áreas de proteção simétricas na rotação

Para áreas de proteção simétricas na rotação (p. ex. mandril do fuso) deve ser descrito o contorno inteiro (não apenas até o centro da rotação!).

Áreas de proteção relativas à ferramenta

Áreas de proteção relativas à ferramenta sempre devem ser convexas. Se for desejada uma área de proteção côncava, esta deve ser distribuída em várias áreas de proteção convexas.



3.2 Ativar/desativar áreas de proteção (CPROT, NPROT)

Função

Ativação ou pré-ativação das áreas de proteção definidas anteriormente para monitoração de colisão ou desativação das áreas de proteção.

O número máximo de áreas de proteção ativas simultaneamente em um canal é definida no dado da máquina.

Se nenhuma área de proteção relativa à ferramenta estiver ativa, então o percurso da ferramenta é verificado para a área de proteção relativa à peça de trabalho.

Indicação

Se nenhuma área de proteção relativa à peça de trabalho estiver ativa, então não haverá nenhuma monitoração da área de proteção.

Sintaxe

```
CPROT(<n>,<state>,<xMov>,<yMov>,<zMov>)
NPROT(<n>,<state>,<xMov>,<yMov>,<zMov>)
```

Significado

`CPROT`:	Chamada da área de proteção específica de canal (apenas para NCU 572/573)
`NPROT`:	Chamada da área de proteção específica da máquina
`<n>`:	Número da área de proteção
`<state>`:	Indicação do estado 0: Desativação da área de proteção 1: Pré-ativação da área de proteção 2: Ativação da área de proteção 3: Pré-ativação da área de proteção com parada condicional
`<xMov>,<yMov>,<zMov>`:	Deslocamento da área de proteção já definida nos eixos geométricos

Condições gerais

Monitoração da área de proteção com correção do raio da ferramenta ativada

Com a correção do raio da ferramenta ativada somente é possível uma monitoração da área de proteção que funcione, se o plano da correção do raio da ferramenta for idêntico ao plano das definições da área de proteção.

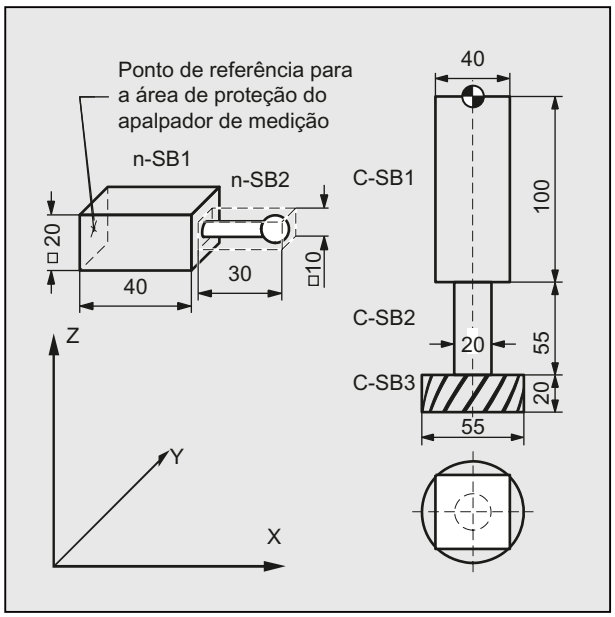
Exemplo

Para uma fresadora deve ser monitorada uma possível colisão da fresa com o apalpador de medição. A posição do apalpador de medição deve ser especificado por um deslocamento para a ativação. Para isso são definidas as seguintes áreas de proteção:

- Uma área de proteção específica da máquina e uma relativa à peça de trabalho para o suporte do apalpador de medição (n-SB1) e para o próprio apalpador de medição (n-SB2).
- Uma área de proteção específica de canal e uma relativa à ferramenta para o porta-fresa (c-SB1), a haste da fresa (c-SB2) e para a própria fresa (c-SB3).

A orientação de todas áreas de proteção está no sentido Z.

A posição do ponto de referência do apalpador de medição para a ativação deve estar em X = -120, Y = 60 e Z = 80.



Código de programa	**Comentário**
DEF INT PROTECAOB	; Definição de uma variável auxiliar
Definição das áreas de proteção do G17	; Ajuste da orientação
NPROTDEF(1,FALSE,3,10,-10)G01 X0 Y-10	; Área de proteção n-SB1
X40	
Y10	
X0	
Y-10	
EXECUTE(PROTECAOB)	
NPROTDEF(2,FALSE,3,5,-5)	; Área de proteção n-SB2
G01 X40 Y-5	
X70	
Y5	
X40	

Código de programa	**Comentário**
Y-5	
EXECUTE(PROTECAOB)	
CPROTDEF(1,TRUE,3,0,-100)	; Área de proteção c-SB1
G01 X-20 Y-20	
X20	
Y20	
X-20	
Y-20	
EXECUTE(PROTECAOB)	
CPROTDEF(2,TRUE,3,-100,-150)	; Área de proteção c-SB2
G01 X0 Y-10	
G03 X0 Y10 J10	
X0 Y-10 J-10	
EXECUTE(PROTECAOB)	
CPROTDEF(3,TRUE,3,-150,-170)	; Área de proteção c-SB3
G01 X0 Y-27,5	
G03 X0 Y27,5 J27,5	
X0 Y27,5 J-27,5	
EXECUTE(PROTECAOB)	
Ativação das áreas de proteção:	
NPROT(1,2,-120,60,80)	; Ativação da área de proteção n-SB1 com deslocamento
NPROT(2,2,-120,60,80)	; Ativação da área de proteção n-SB2 com deslocamento
CPROT(1,2,0,0,0)	; Ativação da área de proteção c-SB1 com deslocamento
CPROT(2,2,0,0,0)	; Ativação da área de proteção c-SB2 com deslocamento
CPROT(3,2,0,0,0)	; Ativação da área de proteção c-SB3 com deslocamento

Outras informações

Estado de ativação (<state>)

- **<state>=2**

  Normalmente uma área de proteção é ativada no programa de peça com estado = 2.

  O estado sempre é específico de canal, mesmo nas áreas de proteção específicas da máquina.

- **<state>=1**

  Se através do programa de usuário PLC estiver previsto que uma área de proteção pode ser ativada pelo programa de usuário PLC, então será realizada a pré-ativação necessária com o estado = 1.

- **<state>=3**

  Na pré-ativação com parada condicional nem sempre ocorre a parada antes de uma área de proteção violada, pré-ativada. A parada somente ocorre se a área de proteção for ativada antes. Isto possibilita uma usinagem sem interrupções, quando as áreas de proteção somente são ativadas em casos especiais. Aqui se deve observar que, em função da rampa de frenagem, um movimento pode invadir a área de proteção, caso a área de proteção tiver sido ativada imediatamente antes da execução.

  A pré-ativação com parada condicional ocorre através do estado = 3.

- **<state>=0**

  A desativação e com isso a desabilitação das áreas de proteção é realizada através do estado = 0. Neste caso não é necessário nenhum deslocamento.

Deslocamento de áreas de proteção na (pré-)ativação

O deslocamento pode ser realizado em 1, 2 ou 3 dimensões. A indicação do deslocamento está em função:

- do ponto zero da máquina para áreas de proteção específicas de peça de trabalho.
- do ponto de referência do porta-ferramenta F para áreas de proteção específicas de ferramenta.

Estado após a inicialização

As áreas de proteção já podem ser ativadas após a inicialização e subsequente aproximação do ponto de referência. Deve-se definir a variável de sistema $SN_PA_ACTIV_IMMED[<n>] ou $SC_PA_ACTIV_IMMED[<n>] como TRUE. Elas sempre são ativadas com o estado = 2 e não possuem deslocamento.

Ativação múltipla de áreas de proteção

Uma área de proteção também pode ser ativada simultaneamente em vários canais (p. ex. cabeçote em duas unidades de avanço dispostas de forma oposta). A monitoração das áreas de proteção somente será realizada quando todos eixos geométricos estiverem referenciados.

Aqui vale:

- A área de proteção não pode ser ativada simultaneamente de forma múltipla com diferentes deslocamentos em um canal.
- Áreas de proteção relativas à máquina devem ter a mesma orientação em ambos canais.

3.3 Verificação quanto à violação da área de proteção, limite da área de trabalho e limites de software (CALCPOSI)

Função

A função CALCPOSI serve para verificar se os eixos geométricos podem percorrer um curso a partir de um ponto de partida especificado sem violar limites de eixo (limites de software), limitações da área de trabalho ou áreas de proteção.

Para o caso em que o curso especificado não pode ser percorrido, será retornado o valor máximo admissível.

A função CALCPOSI é uma subrotina pré-definida. Ela precisa permanecer sozinha em um bloco.

Sintaxe

```
Estado=CALCPOSI(_STARTPOS, _MOVDIST, _DLIMIT, _MAXDIST, _BASE_SYS, _TESTLIM)
```

Significado

Estado	0: Função o. k., o percurso especificado pode ser executado totalmente. –: No _DLIMIT existe pelo menos um componente negativo –: Ocorreu um erro em um cálculo de transformação Se o curso especificado não pode ser percorrido completamente, é retornado um valor decimal positivo: **Posição da unidade (tipo do limite violado):** 1: Os limites de software restringem o percurso. 2: O limite da área de trabalho restringe o percurso. 3: As áreas de proteção restringem o percurso. Se forem violados vários limites simultaneamente (p. ex. limites de software e áreas de proteção), na posição da unidade será mencionado o limite que causa a maior restrição do percurso especificado. **Posição da dezena** 10: O valor inicial viola o limite 20: A reta especificada viola o limite. Este valor também é retornado quando o próprio ponto final não viola nenhum limite, mas durante o percurso do ponto inicial até o ponto final ocorre uma violação do valor limite (p. ex. sair de uma área de proteção, limites de software curvados no WCS com transformações não lineares, p. ex. Transmit).

Posição da centena

100:

O valor limite positivo foi violado (apenas se a posição da unidade for 1 ou 2, isto é, para limites de software e limite da área de trabalho).

100:

Uma área de proteção NCK foi violada (apenas se a posição da unidade for 3).

200:

O valor limite negativo foi violado (apenas se a posição da unidade for 1 ou 2, isto é, para limites de software e limite da área de trabalho).

200:

Uma área de proteção específica de canal foi violada (apenas se a posição da unidade for 3).

Posição da milhar

1000:

Fator com o qual é multiplicado o número do eixo que viola o limite (apenas se a posição da unidade for 1 ou 2, isto é, para limites de software e limite da área de trabalho).

A contagem dos eixos começa com 1 e refere-se, no caso dos limites de software (posição da unidade = 1) violados aos eixos da máquina e, no caso do limite da área de trabalho violada (posição da unidade = 2) aos eixos geométricos.

1000:

Fator com o qual é multiplicado o número da área de proteção violada (apenas se a posição da unidade for 3).

Se várias áreas de proteção foram violadas, é indicada nas posições da centena e da milhar a área de proteção que tem a maior restrição para o percurso especificado.

`_STARTPOS` Valor inicial para abscissa [0], ordenada [1] e terceira coordenada [2] no (WCS)

`_MOVEDIST` Definição de curso incremental para abscissa [0], ordenada [1] e terceira coordenada [2]

`_DLIMIT` [0] - [2]: Distâncias mínimas atribuídas para os eixos geométricos.

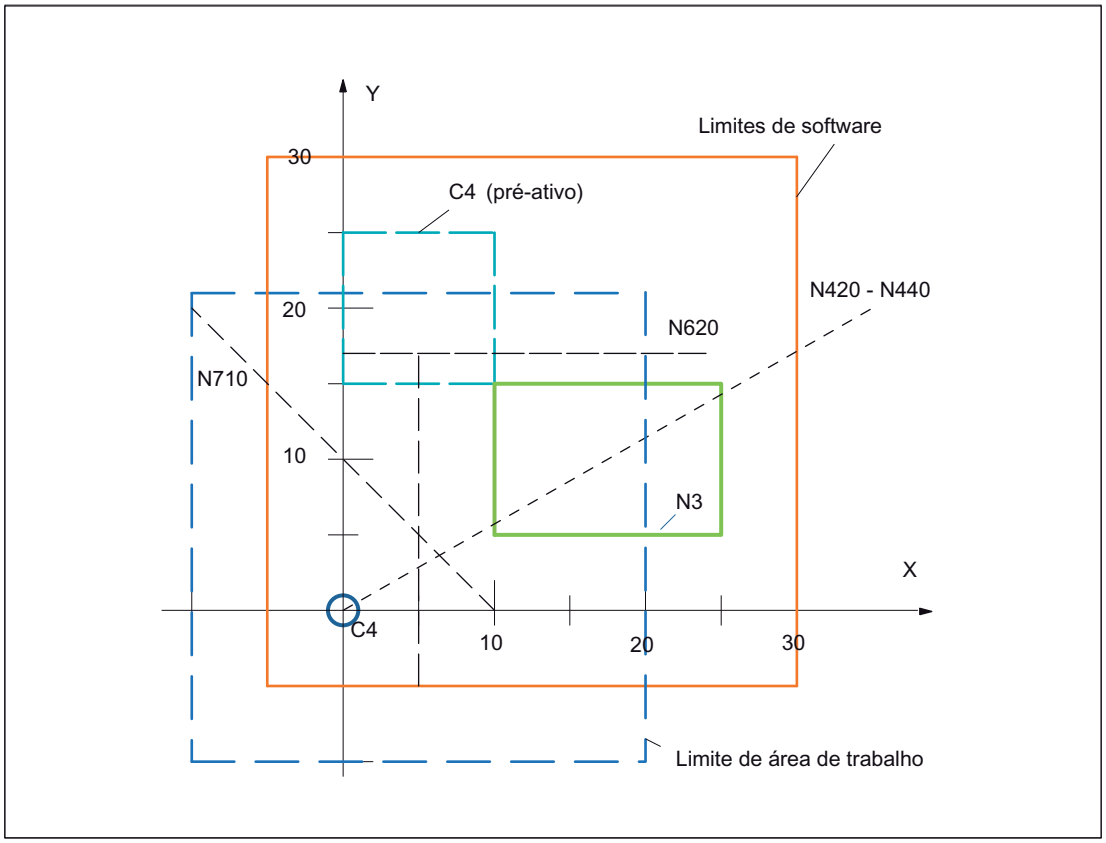
[3]: Distância mínima atribuída para um eixo linear de máquina em uma transformação não linear, se nenhum eixo geométrico foi atribuído como único.

[4]: Distância mínima atribuída para um eixo rotativo de máquina em uma transformação não linear, se nenhum eixo geométrico foi atribuído como único. Apenas para transformações especiais onde devem ser monitorados os limites de software.

_MAXDIST	Campo [0] - [2] para valor de retorno. Curso incremental em todos três eixos geométricos, sem que a distância mínima especificada não seja alcançada por um limite de eixo nos eixos envolvidos. Se o percurso não está limitado, então o conteúdo deste parâmetro de retorno é igual ao do _MOVDIST.
_BASE_SYS	FALSE ou parâmetro não especificado: Para a avaliação das especificações de posição e comprimento é feita a avaliação do código G do grupo 13 (G70, G71, G700, G710; polegadas/métrico). Com o G70 ativo e se o sistema básico for métrico (ou G71 ativo e o sistema básico for em polegadas), as variáveis de sistema relativas ao WCS, $AA_IW[X] e $AA_MW[X]), são enviadas para o sistema básico e, se necessário, deverão ser convertidas para serem utilizadas na função CALCPOSI. TRUE: Para a avaliação das especificações de posição e comprimento, sempre é utilizado o sistema básico do comando, independente do valor ativo do código G do grupo 13.
_TESTLIM	Limitações para verificação (código binário): 1: Monitorar limites de software 2: Monitorar limites da área de trabalho 3: Monitorar áreas de proteção ativadas 4: Monitorar áreas de proteção pré-ativadas Combinações pela adição dos valores. Padrão: 15; verificar todas.

Exemplo

No exemplo (figura) estão indicados os limites de software e limites da área de trabalho no plano X-Y. Adicionalmente foram definidas três áreas de proteção, as duas áreas específicas de canal C2 e C4 assim como a área específica NCK, a N3. C2 é uma área de proteção circular, ativa, relativa à ferramenta, com 2 mm de raio. C4 é uma área de proteção quadrada, pré-ativada, relativa à peça de trabalho, com 10 mm de laterais e a N3 é uma área de proteção retangular, ativa, com 10 mm e 15 mm de laterais. No seguinte programa NC são definidas primeiramente as áreas de proteção e as limitações da área de trabalho como no esboço, depois é chamada a função CALCPOSI com diversas parametrizações. Os resultados de cada chamada do CALCPOSI estão resumidos na tabela no fim do exemplo.



Código de programa	Comentário
N10 def real _STARTPOS[3]	
N20 def real _MOVDIST[3]	
N30 def real _DLIMIT[5]	
N40 def real _MAXDIST[3]	
N50 def int _SB	
N60 def int _STATUS	

Código de programa	**Comentário**
N70 cprotdef(2, true, 0)	; área de proteção relativa à ferramenta
N80 g17 g1 x-y0	
N90 g3 i2 x2	
N100 i-x-	
N110 execute(_SB)	
N120 cprotdef(4, false, 0)	; área de proteção relativa à peça de trabalho
N130 g17 g1 x0 y15	
N140 x10	
N150 y25	
N160 x0	
N170 y15	
N180 execute(_SB)	
N190 nprotdef(3, false, 0)	; área de proteção relativa à máquina
N200 g17 g1 x10 y5	
N210 x25	
N220 y15	
N230 x10	
N240 y5	
N250 execute(_SB)	
N260 cprot(2,2,0, 0, 0)	; ; ativar ou pré-ativar áreas de proteção
N270 cprot(4,1,0, 0, 0)	
N280 nprot(3,2,0, 0, 0)	
N290 g25 XX=-YY=-	; Definir limites da área de trabalho
N300 g26 xx= 20 yy= 21	
N310 _STARTPOS[0] = 0.	
N320 _STARTPOS[1] = 0.	
N330 _STARTPOS[2] = 0.	
N340 _MOVDIST[0] = 35.	
N350 _MOVDIST[1] = 20.	
N360 _MOVDIST[2] = 0.	
N370 _DLIMIT[0] = 0.	
N380 _DLIMIT[1] = 0.	
N390 _DLIMIT[2] = 0.	
N400 _DLIMIT[3] = 0.	
N410 _DLIMIT[4] = 0.	
;Diversas chamadas de funções	; Outro ponto de partida
N420 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST)	
N430 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST,,3)	
N440 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST,,1)	

Código de programa	**Comentário**
N450 _STARTPOS[0] = 5.	; Outro destino
N460 _STARTPOS[1] = 17.	
N470 _STARTPOS[2] = 0.	
N480 _MOVDIST[0] = 0.	
N490 _MOVDIST[1] =-.	
N500 _MOVDIST[2] = 0.	
;Diversas chamadas de funções	
N510 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST,,14)	
N520 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST,, 6)	
N530 _DLIMIT[1] = 2.	
N540 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST,, 6)	
N550 _STARTPOS[0] = 27.	
N560 _STARTPOS[1] = 17.1	
N570 _STARTPOS[2] = 0.	
N580 _MOVDIST[0] =-.	
N590 _MOVDIST[1] = 0.	
N600 _MOVDIST[2] = 0.	
N610 _DLIMIT[3] = 2.	
N620 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST,, 12)	
N630 _STARTPOS[0] = 0.	
N640 _STARTPOS[1] = 0.	
N650 _STARTPOS[2] = 0.	
N660 _MOVDIST[0] = 0.	
N670 _MOVDIST[1] = 30.	
N680 _MOVDIST[2] = 0.	
N690 trans x10	
N700 arot z45	
N710 _STATUS = calcposi(_STARTPOS,_MOVDIST, _DLIMIT, _MAXDIST)	
N720 M30	

Resultados das verificações no exemplo:

Bloco nº N...	_STATUS	_MAXDIST [0] (= X)	_MAXDIST [1] (= Y)	Observações
420	3123	8.040	4.594	Área de proteção SB N3 é violada.
430	1122	20.000	11.429	Não é violada nenhuma monitoração da área de proteção e limite de área de trabalho.
440	1121	30.000	17.143	Apenas a monitoração dos limites de software está ativa.
510	4213	0.000	0.000	Ponto de partida viola a área de proteção C4
520	0000	0.000	–.000	Área de proteção C4 pré-ativada não é monitorada. Curso especificado pode ser percorrido completamente.
540	2222	0.000	–.000	O percurso é restrito pelo limite da área de trabalho devido o _DLIMIT[1]=2.
620	4223	–.000	0.000	A distância para C4 é de 4 mm ao total devido o C2 e _DLIMIT[3]. A distância C2 –N3 de 0.1 mm não limita o percurso.
710	1221	0.000	21.213	Frame ativo com translação e rotação. O percurso permitido em _MOVDIST é válido no sistema de coordenadas (WCS) movido e girado.

Casos especiais e outros detalhes

Todos os dados de curso sempre são especificados em raio, mesmo para um eixo transversal com código G "DIAMON" ativo. Se o curso não pode ser percorrido completamente pelo eixo envolvido, no valor de retorno _MAXDIST os cursos dos outros eixos também são reduzidos proporcionalmente, de modo que o ponto final resultante esteja na trajetória especificada.

É permitido que não sejam definidos limites de software e limites da área de trabalho ou áreas de proteção para um ou vários dos eixos envolvidos. Todos limites somente serão monitorados se os eixos envolvidos estiverem referenciados. Qualquer eixo rotativo envolvido somente será monitorado se não for um eixo Modulo.

A monitoração dos limites de software e dos limites de área de trabalho é realizada como no percurso normal, em função dos ajustes ativos (sinais de interface para seleção do limite de software 1 ou 2, código G WALIMON / WALIMOF, dados de ajuste para ativação individual dos limites da área de trabalho e para definir se a monitoração dos limites da área de trabalho deve considerar o raio da ferramenta ativa ou não).

Em determinadas transformações cinemáticas (p. ex. TRANSMIT) a posição dos eixos da máquina não pode ser definida como única a partir das posições do sistema de coordenadas da peça de trabalho (WCS). Em um percurso normal a condição inequívoca normalmente resulta do histórico e da condição de que um movimento contínuo em WCS deve corresponder a um movimento contínuo dos eixos da máquina. Por isso que nestes casos, na monitoração dos limites de software com o auxílio da função CALCPOSI, se chama a atual posição da máquina para desfazer a ambigüidade. Eventualmente se deve programar um **STOPRE** antes do CALCPOSI, para que a função possa ser atribuída com posições válidas dos eixos da máquina.

Não se pode assegurar que nas áreas de proteção sempre seja mantida a mesma distância especificada no _DLIMIT[3] durante um movimento no percurso indicado. Dessa forma, em caso de prolongamento do ponto final retornado no _MOVDIST não pode ocorrer nenhuma violação de área de proteção nesta distância. Porém, a linha reta pode percorrer seu percurso bem próxima a uma área de proteção.

Indicação

Mais detalhes sobre limites de campos de trabalho estão disponíveis no
/PG/ Manual de programação Fundamentos,

e sobre os limites de software, no
/FB1/ Manual de funções básicas; Monitorações de eixos, Áreas de proteção (A3).

Comandos especiais de curso 4

4.1 Aproximar posições codificadas (CAC, CIC, CDC, CACP, CACN)

Função

Através dos seguintes comandos podemos deslocar eixos lineares e rotativos através de números de posição que estão registrados como posições fixas de eixo em tabelas de dados da máquina. Este tipo de programação é denominado de "Aproximação de posições codificadas".

Sintaxe

```
CAC(<n>)
CIC(<n>)
CACP(<n>)
CACN(<n>)
```

Significado

`CAC(<n>)`	Aproximar posição codificada de número de posição n
`CIC(<n>)`	Aproximar a posição codificada, partindo-se do atual número de posição, n locais de posição para frente (+n) ou para trás (–n)
`CDC(<n>)`	Aproximar posição codificada de número de posição n pelo percurso mais curto (apenas para eixos rotativos)
`CACP(<n>)`	Aproximar posição codificada de número de posição n em sentido positivo (apenas para eixos rotativos)
`CACN(<n>)`	Aproximar posição codificada de número de posição n em sentido negativo (apenas para eixos rotativos)
`<n>`	Número de posição dentro da tabela de dados da máquina Faixa de valores: 0, 1, … (número máx. de locais na tabela - 1)

Exemplo: Aproximação de posições codificadas de um eixo de posicionamento

```
Código de programação      Comentário
N10 FA[B]=300              ; Avanço para eixo de posicionamento B
N20 POS[B]=CAC(10)         ; Aproximar a posição codificada de número de posição 10
N30 POS[B]=CIC(-4)         ; Aproximar a posição codificada do "atual número de
                             posição" - 4
```

Literatura

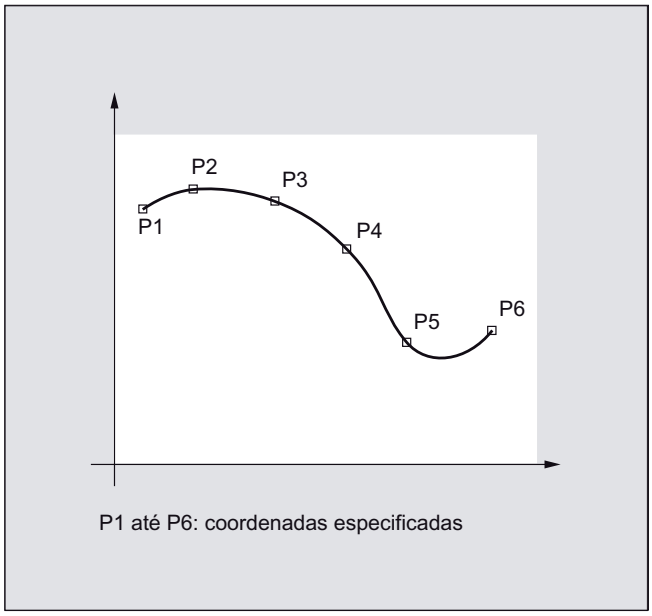
- Manual de funções ampliadas; Eixos de indexação (T1)
- Manual de funções para ações sincronizadas

4.2 Interpolação de Spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL)

Função

Muitos contornos curvados nas peças de trabalho não podem ser descritos analiticamente com precisão. Por isso que os contornos desse tipo são aproximados através de um número limitado de pontos de interpolação, como no caso da digitalização de superfícies. Para gerar a superfície digitalizada de uma peça de trabalho os pontos de interpolação devem ser ligados em uma descrição de contorno. Isso permite a interpolação de Spline.

Uma Spline define uma curva formada por polinômios de 2º e 3º grau. As propriedades nos pontos de interpolação de uma Splines podem ser definidas **em função do tipo de Spline utilizado**.



P1 até P6: coordenadas especificadas

No SINUMERIK solution line estão disponíveis os seguintes tipos de Spline:

- A-Spline
- B-Spline
- C-Spline

Sintaxe

Geral:

```
ASPLINE X... Y... Z... A... B... C...
BSPLINE X... Y... Z... A... B... C...
CSPLINE X... Y... Z... A... B... C...
```

Na B-Spline também se programa:

```
PW=<n>
SD=2
PL=<valor>
```

Na A-Spline e C-Spline também se programa:

`BAUTO` / `BNAT` / `BTAN`

`EAUTO` / `ENAT` / `ETAN`

Significado

Tipo de interpolação de Spline:

`ASPLINE`	Comando para ativar a interpolação de A-Spline
`BSPLINE`	Comando para ativar a interpolação de B-Spline
`CSPLINE`	Comando para ativar a interpolação de C-Spline
	Os comandos `ASPLINE`, `BSPLINE` e `CSPLINE` são ativados modalmente e pertencem ao grupo dos comandos de curso.

Pontos de interpolação e pontos de controle:

`X... Y... Z...` `A... B... C...`	Posições em coordenadas cartesianas

Peso de ponto (apenas para B-Spline):

`PW`	Com o comando `PW` é possível ser programado um chamado "Peso de ponto" para cada ponto de interpolação.		
`<n>`	"Peso de ponto"		
	Faixa de valores:	0 ≤ n ≤ 3	
	Incremento:	0.0001	
	Efeito:	n > 1	A curva é guiada de forma mais acentuada pelo ponto de controle.
		n < 1	A curva é guiada de forma menos acentuada pelo ponto de controle

Grau Spline (apenas para B-Spline):

`SD`	Como padrão é usado um polinômio de 3º grau. Através da programação do `SD=2`, mas também pode ser utilizado um polinômio de 2º grau.

Distância dos nós (apenas para B-Spline):

`PL` As distâncias entre os nós são calculadas internamente de forma adequada. O comando também pode processar as distâncias indicadas dos nós, que são indicadas como distância parâmetro-intervalo com o comando `PL`.

`<valor>` Comprimento de intervalo do parâmetro

Faixa de valores: como dimensão de curso

Comportamento de transição no início da curva Spline (apenas para A-Spline ou C-Spline):

`BAUTO` Nenhuma especificação para comportamento de transição. O início resulta da posição do primeiro ponto.

`BNAT` Curvatura zero

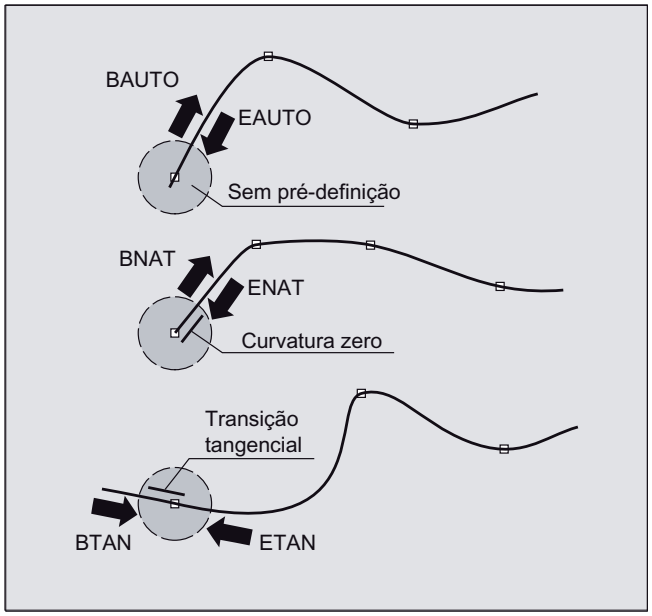
`BTAN` Transição tangencial para o bloco anterior (ajuste inicial)

Comportamento de transição no fim da curva Spline (apenas para A-Spline ou C-Spline):

`EAUTO` Nenhuma especificação para comportamento de transição. O fim resulta da posição do último ponto.

`ENAT` Curvatura zero

`ETAN` Transição tangencial para o bloco anterior (ajuste inicial)



Indicação

O comportamento de transição programável não exerce nenhuma influência sobre a B-Spline. A B-Spline sempre é tangencial ao polígono de controle no ponto de partida e ponto final.

Condições gerais

- A correção do raio da ferramenta é aplicável.
- A monitoração de colisões é realizada na projeção sobre o plano.

Exemplos

Exemplo 1: B-Spline

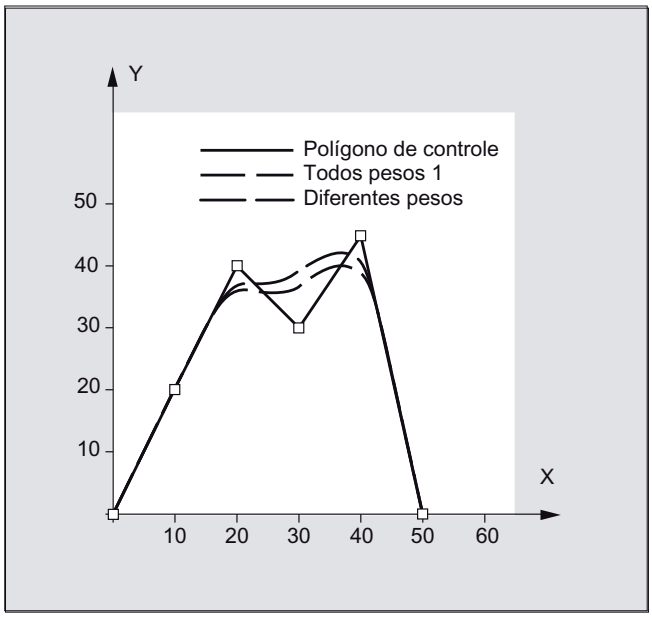
Código de programa 1 (todos os pesos 1)

```
N10 G1 X0 Y0 F300 G64
N20 BSPLINE
N30 X10 Y20
N40 X20 Y40
N50 X30 Y30
N60 X40 Y45
N70 X50 Y0
```

Código de programa 2 (diferentes pesos)

```
N10 G1 X0 Y0 F300 G64
N20 BSPLINE
N30 X10 Y20 PW=2
N40 X20 Y40
N50 X30 Y30 PW=0.5
N60 X40 Y45
N70 X50 Y0
```

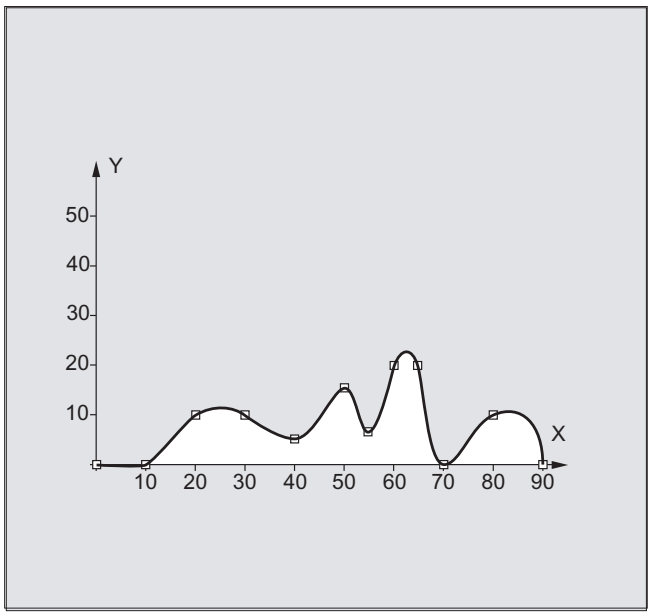
Código de programa 3 (polígono de controle)	**Comentário**
N10 G1 X0 Y0 F300 G64	
N20	; não aplicado
N30 X10 Y20	
N40 X20 Y40	
N50 X30 Y30	
N60 X40 Y45	
N70 X50 Y0	



Exemplo 2: C-Spline, no início e no fim da curvatura zero

Código de programa

```
N10 G1 X0 Y0 F300
N15 X10
N20 BNAT ENAT
N30 CSPLINE X20 Y10
N40 X30
N50 X40 Y5
N60 X50 Y15
N70 X55 Y7
N80 X60 Y20
N90 X65 Y20
N100 X70 Y0
N110 X80 Y10
N120 X90 Y0
N130 M30
```



Exemplo 3: Interpolação de Spline (A-Spline) e transformação de coordenadas (ROT)

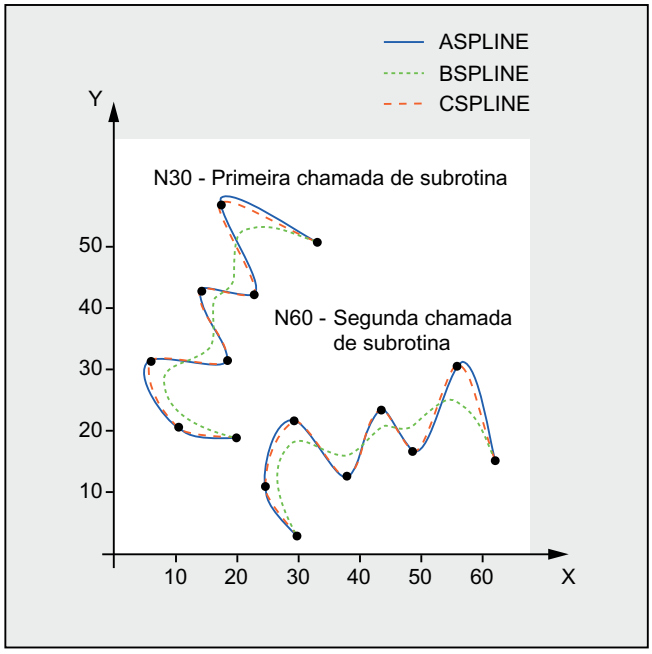
Programa principal:

Código de programa	Comentário
N10 G00 X20 Y18 F300 G64	; Aproximar o ponto de partida.
N20 ASPLINE	; Ativar o tipo de interpolação A-Spline.
N30 CONTORNO	; Primeira chamada da subrotina.
N40 ROT Z-45	; Transformação de coordenadas: Rotação do WCS em -45° em torno do eixo Z.
N50 G00 X20 Y18	; Aproximar o ponto de partida do contorno.
N60 CONTORNO	; Segunda chamada da subrotina.
N70 M30	; Fim do programa

Subrotina "Contorno" (contém as coordenadas dos pontos de interpolação):

Código de programa
N10 X20 Y18
N20 X10 Y21
N30 X6 Y31
N40 X18 Y31
N50 X13 Y43
N60 X22 Y42
N70 X16 Y58
N80 X33 Y51
N90 M1

Na figura a seguir, além da curva Spline, resultante do exemplo de programa (ASPLINE), também estão contidas as curvas Spline que resultariam da ativação de uma interpolação B-Spline ou C-Spline (BSPLINE, CSPLINE):



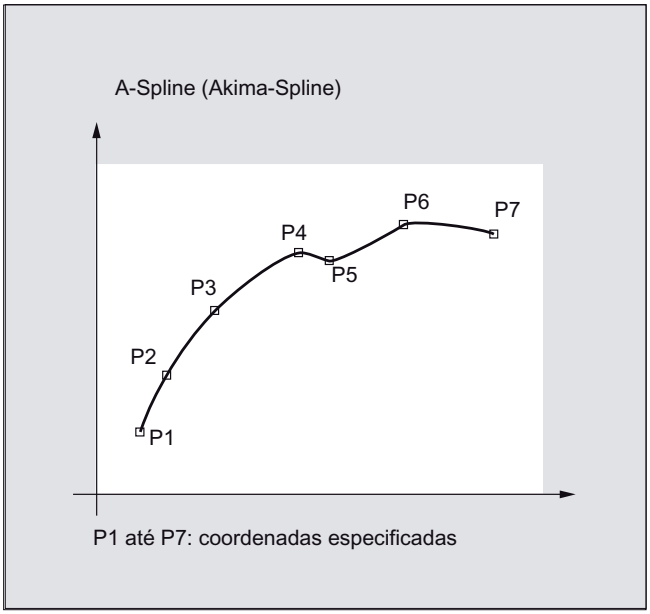
Outras informações

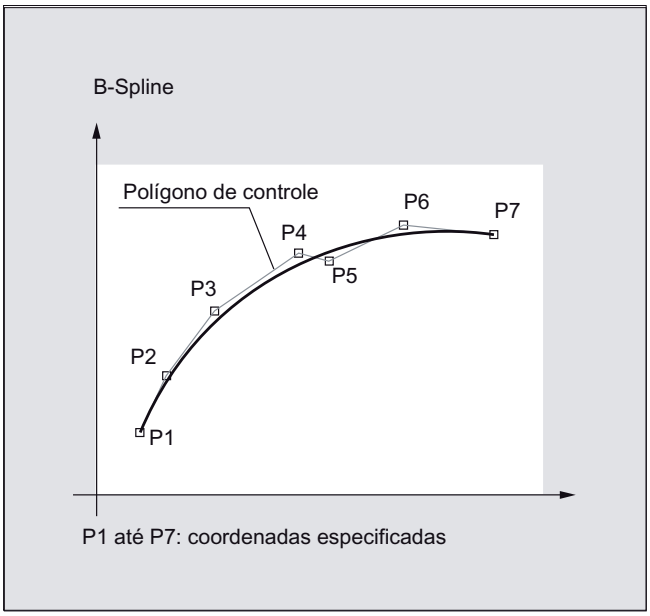
Vantagens da interpolação de Spline

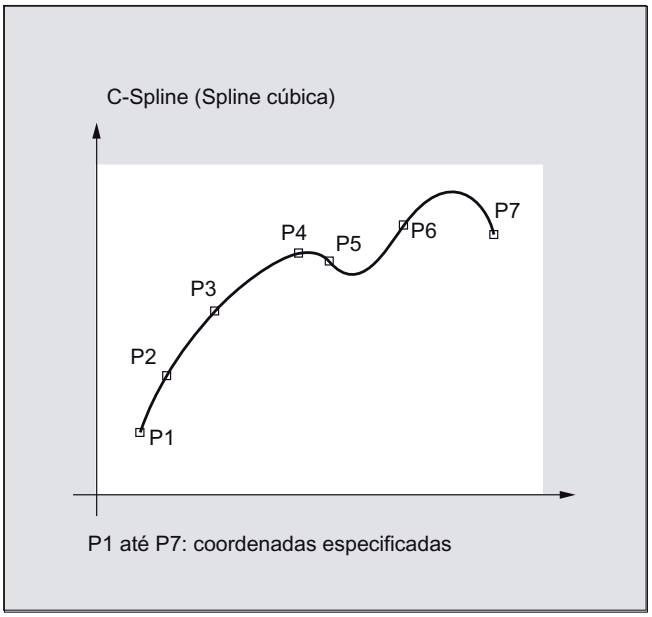
Através do uso da interpolação de Spline são obtidas as vantagens mencionadas a seguir, em contrapartida ao uso de blocos de retas `G01`:

- Redução do número de blocos de programa de peça necessários para descrição do contorno
- Trajetória suave de curvas e de melhor aspecto estético nas transições entre os blocos de programa de peça

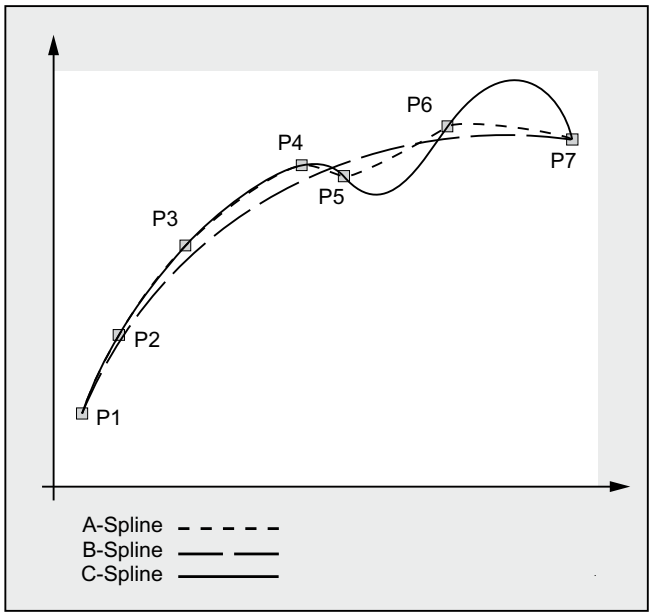
Propriedades e aplicação dos diversos tipos de Spline

Tipo de Spline	Propriedades e aplicação
A-Spline	 **Propriedades:** • Percorre exatamente pelos pontos de interpolação especificados. • A trajetória da curva é tangencial, mas nem sempre curvada. • Quase não gera nenhuma ondulação indesejada. • A área de influência de alterações de pontos de interpolação é local, isto é, a alteração de um ponto de interpolação somente tem efeito até 6 pontos de interpolação vizinhos. **Aplicação:** A A-Spline é principalmente indicada para interpolação de trajetórias de curvas com grandes variações de inclinação (p. ex. trajetórias de curvas em forma de escada).

Tipo de Spline	Propriedades e aplicação
B-Spline	 **Propriedades:** • Não percorre pelos pontos de interpolação especificados, mas apenas na proximidade dos mesmos. As curvas são direcionadas pelos pontos de interpolação. Também se pode controlar a trajetória de curvas através da avaliação de peso dos pontos de interpolação com um fator. • A trajetória de curvas é tangencial e sempre curvada. • Não gera nenhuma ondulação indesejada. • A área de influência de alterações de pontos de interpolação é local, isto é, a alteração de um ponto de interpolação somente tem efeito até 6 pontos de interpolação vizinhos. **Aplicação:** A princípio o B-Spline é indicado como interface para sistemas CAD.

Tipo de Spline	Propriedades e aplicação
C-Spline	 **Propriedades:** • Percorre exatamente pelos pontos de interpolação especificados. • A trajetória de curvas é tangencial e sempre curvada. • Gera ondulações indesejadas com freqüência, principalmente em pontos com grandes variações de inclinação. • A área de influência das variações de pontos de interpolação é global, isto é, a variação de um ponto de interpolação atua pela trajetória inteira da curva. **Aplicação:** O C-Spline pode ser bem empregado se os pontos de interpolação estiverem em uma curva analítica conhecida (círculo, parábola, hipérbole)

Comparação dos três tipos de Spline usando os mesmos pontos de interpolação



Número mínimo de blocos de Spline

Os códigos G ASPLINE, BSPLINE e CSPLINE ligam os pontos de fim de bloco com Splines. Para isso, antecipadamente, deve ser calculada simultaneamente uma série de blocos (pontos finais). O tamanho padrão do buffer para os cálculos é de 10 blocos. Nem toda informação de bloco é um ponto final de Spline. Entretanto, o comando requer de 10 blocos uma determinada quantidade de blocos de ponto final Spline:

Tipo de Spline	Número mínimo de blocos de Spline
A-Spline:	De cada 10 blocos, no mínimo **4** devem ser blocos Spline. Os blocos de comentário e cálculos de parâmetros não são considerados neste caso.
B-Spline:	De cada 10 blocos, no mínimo **6** devem ser blocos Spline. Os blocos de comentário e cálculos de parâmetros não são considerados neste caso.
C-Spline:	A quantidade mínima requerida em blocos Spline resulta da seguinte somatória: Valor do MD20160 $MC_CUBIC_SPLINE_BLOCKS + 1 No MD20160 se especifica a quantidade de pontos através dos quais é calculado o segmento de Spline. O ajuste padrão é 8. Por isso que, em caso padrão, de cada 10 blocos pelo menos **9** devem ser blocos Spline.

Indicação

Se o valor ficar abaixo do tolerável será emitido um alarme, e o mesmo ocorre se um eixo envolvido com a Spline for programado como eixo de posicionamento.

Agrupamento de blocos Spline curtos

Na interpolação de Spline podem existir blocos Spline curtos, que causam uma redução desnecessária da velocidade de percurso. Com a função "Agrupamento de blocos Spline curtos" estes blocos podem ser agrupados de modo que o tamanho resultante dos blocos seja suficiente e que não resulte em uma redução da velocidade de percurso.

A função é ativada através do dado de máquina específico de canal:

MD20488 $MC_SPLINE_MODE (ajuste da interpolação de Spline)

Literatura:
Manual de funções básicas; Modo de controle da trajetória, Parada exata, LookAhead (B1), Capítulo: Agrupamento de blocos Spline curtos

4.3 Agrupamento de Spline (SPLINEPATH)

Função

Os eixos que são interpolados no agrupamento de Spline são selecionados com o comando SPLINEPATH. A interpolação de Splines é possível até com oito eixos.

Indicação

Se o SPLINEPATH não for programado de forma explícita, então são deslocados os primeiros três eixos do canal do agrupamento de Spline.

Sintaxe

A definição do agrupamento de Spline é realizado em um bloco especial:

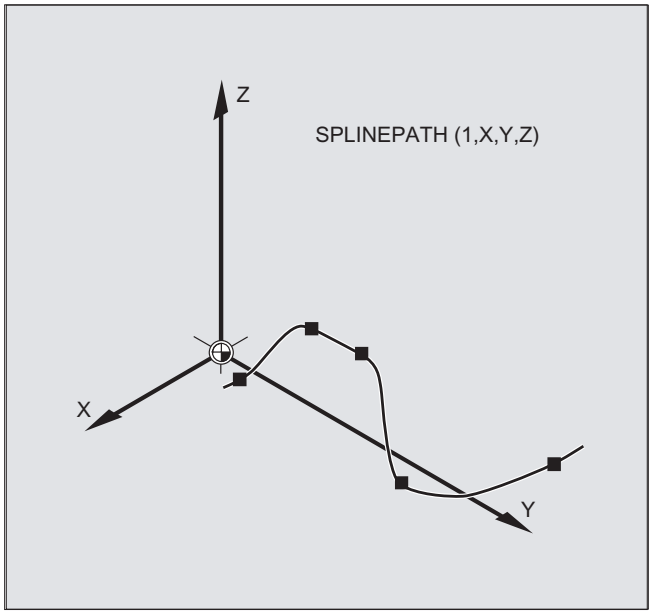
```
SPLINEPATH(n,X,Y,Z,…)
```

Significado

SPLINEPATH	Comando para definição de um agrupamento de Spline
n	=1 (valor fixo)
X,Y,Z,…	Identificador dos eixos de trajetória interpolados no agrupamento de Spline

Exemplo: Agrupamento de Spline com três eixos de percurso

Código de programa	**Comentário**
N10 G1 X10 Y20 Z30 A40 B50 F350	
N11 SPLINEPATH(1,X,Y,Z)	; Agrupamento de Spline
N13 CSPLINE BAUTO EAUTO X20 Y30 Z40 A50 B60	; C-Spline
N14 X30 Y40 Z50 A60 B70	; Pontos de apoio
…	
N100 G1 X… Y…	; Desseleção da interpolação de Spline



4.4 Compressão de blocos NC (COMPON, COMPCURV, COMPCAD, COMPOF)

Função

Sistemas CAD e CAM normalmente fornecem blocos lineares que mantém a precisão parametrizada. Em contornos complexos isso resulta em uma elevada quantidade de dados e eventuais segmentos curtos de percurso. Estes segmentos curtos de percurso limitam a velocidade de processamento.

Através da aplicação de uma função de compressor se realiza uma aproximação do contorno especificado por blocos lineares através de blocos de polinômios. Com isso resultam as seguintes vantagens:

- Redução do número necessário de blocos de programa de peça para descrição do contorno da peça
- Transições constantes de blocos
- Elevação das velocidades de percurso máximas

Estão disponíveis as seguintes funções de compressor:

- COMPON

  As transições de blocos somente são constantes na **velocidade** enquanto a aceleração dos eixos envolvidos permitir a realização de saltos nas transições de blocos.

- COMPCURV

  As transições de blocos são **constantes na aceleração**. Com isso se assegura uma distribuição suave da velocidade e também da aceleração de todos eixos nas transições dos blocos.

- COMPCAD

  Compressão intensiva em tempo de processamento e em espaço de memória que otimiza a qualidade superficial e a velocidade. O COMPCAD somente deve ser utilizado se as medidas do programa CAD/CAM não garantirem a melhora de superfície.

A função de compressor é encerrada com `COMPOF`.

Sintaxe

```
COMPON
COMPCURV
COMPCAD
COMPOF
```

Significado

COMPON:	Comando para ativação da função de compressor COMPON.	
	Efeito:	modal
COMPCURV:	Comando para ativação da função de compressor COMPCURV.	
	Efeito:	modal
COMPCAD:	Comando para ativação da função de compressor COMPCAD.	
	Efeito:	modal
COMPOF:	Comando para desativar a função de compressor atualmente ativa.	

Indicação

Para uma melhoria adicional da qualidade superficial pode ser usada a função de suavização G642 e a limitação de solavancos SOFT. Estes comandos devem ser escritos no início do programa.

Condições gerais

- Geralmente a compressão de blocos NC é executada apenas em blocos lineares (G1).
- Somente são comprimidos os blocos que apresentarem uma sintaxe simples:

  ```
  N... G1X... Y... Z... F... ; Comentário
  ```

  Todos demais blocos são processados sem serem alterados (sem compressão).
- Os blocos de movimento com endereços ampliados como C=100 ou A=AC(100) também são comprimidos.
- Os valores de posição não precisam ser programados diretamente, eles também podem ser especificados indiretamente através de instruções de parâmetros, p. ex. X=R1*(R2+R3).
- Se o opcional "Transformação de orientação" estiver disponível, também podem ser comprimidos os blocos NC onde a orientação de ferramenta (e eventualmente a rotação da ferramenta) estiver programada através de vetores (veja " Compressão da orientação (Página 361) ").
- O processo de compressão é cancelado por qualquer outra instrução NC, como p. ex. a emissão de uma função auxiliar.

Exemplos

Exemplo 1: COMPON

```
Código de programa                    Comentário
N10 COMPON                            ; Função de compressor COMPON ativada.
N11 G1 X0.37 Y2.9 F600                ; G1 antes do ponto final e avanço.
N12 X16.87 Y-.698
N13 X16.865 Y-.72
N14 X16.91 Y-.799
…
N1037 COMPOF                          ; Função de compressor desativada.
…
```

Exemplo 2: COMPCAD

```
Código de programa                    Comentário
G00 X30 Y6 Z40
G1 F10000 G642                        ; Função de suavização G642 ativada.
SOFT                                  ; Limitação de solavancos SOFT ativada.
COMPCAD                               ; Função de compressor COMPCAD ativada.
STOPFIFO
N24050 Z32.499
N24051 X41.365 Z32.500
N24052 X43.115 Z32.497
N24053 X43.365 Z32.477
N24054 X43.556 Z32.449
N24055 X43.818 Z32.387
N24056 X44.076 Z32.300
…
COMPOF                                ; Função de compressor desativada.
G00 Z50
M30
```

Literatura

Manual de funções básicas; Modo de controle da trajetória, Parada exata, LookAhead (B1), Capítulo: "Compressão de blocos NC"

4.5 Interpolação de polinômios (POLY, POLYPATH, PO, PL)

Função

Na verdade trata-se de uma interpolação de polinômios (`POLY`) e não de um tipo de interpolação de Splines. Em primeira linha ela age como interface para a programação de curvas Spline criadas externamente. Aqui os segmentos Spline podem ser programados diretamente.

Este tipo de interpolação alivia o NC dos cálculos dos coeficientes de polinômio. Ela pode perfeitamente ser aplicada se os coeficientes são diretamente fornecidos de um sistema CAD ou pós-processador.

Sintaxe

Polinômio de 3º grau:

```
POLY PO[X]=(xe,a2,a3) PO[Y]=(ye,b2,b3) PO[Z]=(ze,c2,c3) PL=n
```

Polinômio de 5º grau e nova sintaxe de polinômio:

```
POLY X=PO(xe,a2,a3,a4,a5) Y=PO(ye,b2,b3,b4,b5) Z=PO(ze,c2,c3,c4,c5)
PL=n
POLYPATH("AXES","VECT")
```

Indicação

A soma dos coeficientes de polinômio e eixos programados em um bloco NC não pode exceder o número máximo permitido de eixos por bloco.

Significado

`POLY` :	Ativação da interpolação de polinômios com um bloco com POLY.
`POLYPATH` :	A interpolação de polinômios pode ser selecionada para os dois grupos de eixos, AXIS ou VECT
`PO[identificador de eixo/variável]` :	Pontos finais e coeficientes de polinômio
`X, Y, Z` :	Identificador de eixo
`x`e, `ye, ze` :	Especificação da posição final do respectivo eixo; faixa de valores como dimensão de curso

`a2, a3, a4, a5` :	Os coeficientes a_2, a_3, a_4, e a_5 são escritos com seu respectivo valor; faixa de valores como na dimensão de curso. O último coeficiente em cada caso pode ser omitido se o valor for zero.
`PL` :	Comprimento do intervalo de parâmetro no qual está definido o polinômio (faixa de definição da função f(p)). O intervalo sempre inicia em 0, p pode aceitar valores de 0 até PL. Faixa de valores teórica para PL: 0,0001 … 99 999,9999 **Nota:** O valor PL vale para o bloco em que ele se encontra. Se não foi programado PL, então PL=1.

Ativação e desativação da interpolação de polinômios

A interpolação de polinômios é ativada no programa de peça através do comando G `POLY`.

O comando G `POLY` pertence ao 1º grupo G, junto com o `G0`, `G1`, `G2`, `G3`, `ASPLINE`, `BSPLINE` e `CSPLINE`.

Os eixos que foram programados apenas com nomes e ponto final (p. ex. `X10`) são deslocados linearmente. Se todos os eixos de um bloco NC foram programados desta forma, o comando se comporta como no caso do G1.

De modo implícito a interpolação de polinômios é novamente desativada através da programação de outro comando do 1º grupo G (p. ex. `G0`, `G1`).

Coeficiente de polinômio

Os valores `PO (PO[]=)` e `...=PO(...)` especifica todos os coeficientes de polinômio para um eixo. Dependendo do grau do polinômio mais valores são especificados separados por vírgula. Dentro de um bloco são possíveis diferentes graus de polinômio para diferentes eixos.

Subrotina POLYPATH

Com `POLYPATH(...)` é possível liberar a interpolação de polinômios de modo seletivo para determinados grupos de eixos.

Apenas os eixos de percurso e eixos adicionais:	`POLYPATH("AXES")`
Somente eixos de orientação: (no processo com transformação de orientação)	`POLYPATH ("VECT")`

Os eixos que não forem liberados são deslocados linearmente.

Como padrão a interpolação de polinômios é liberada para os dois grupos de eixos.

Através da programação sem indicar o parâmetro `POLYPATH( )` desativa-se a interpolação de polinômios para todos os eixos.

Exemplo

```
Código de programa                                              Comentário
N10 G1 X… Y… Z… F600
N11 POLY PO[X]=(1,2.5,0.7) PO[Y]=(0.3,1,3.2) PL=1.5             ; Interpolação de polinômios ativada
N12 PO[X]=(0,2.5,1.7) PO[Y]=(2.3,1.7) PL=3
...
N20 M8 H126 …
N25 X70 PO[Y]=(9.3,1,7.67) PL=5                                 ; especificações mistas para os eixos
N27 PO[X]=(10,2.5) PO[Y]=(2.3)                                  ; nenhum PL programado; então PL=1
N30 G1 X… Y… Z.                                                 ; Interpolação de polinômios desativada
…
```

Exemplo: Nova sintaxe de polinômio

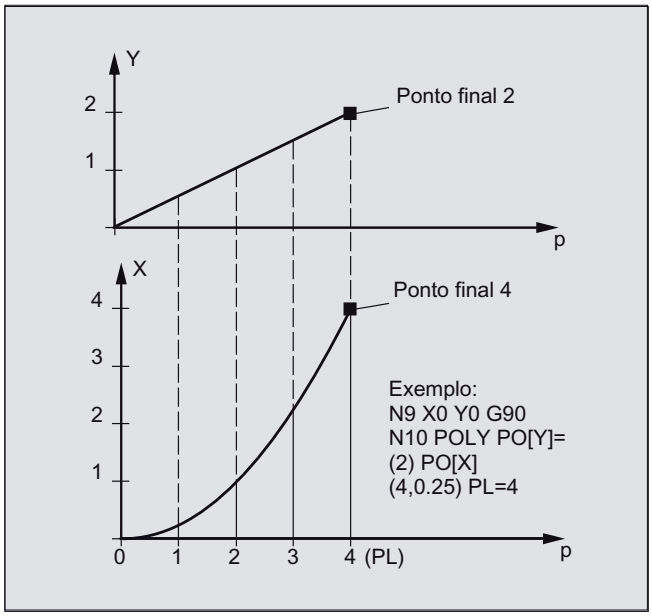
Sintaxe de polinômio ainda aplicada	Nova sintaxe de polinômio
PO[identificador de eixo]=(.. , ..)	Identificador de eixo=PO(.. , ..)
PO[PHI]=(.. , ..)	PHI=PO(.. , ..)
PO[PSI]=(.. , ..)	PSI=PO(.. , ..)
PO[THT]=(.. , ..)	THT=PO(.. , ..)
PO[]=(.. , ..)	PO(.. , ..)
PO[variável]=IC(.. , ..)	variável=PO IC(.. , ..)

Exemplo: Curva no plano X/Y

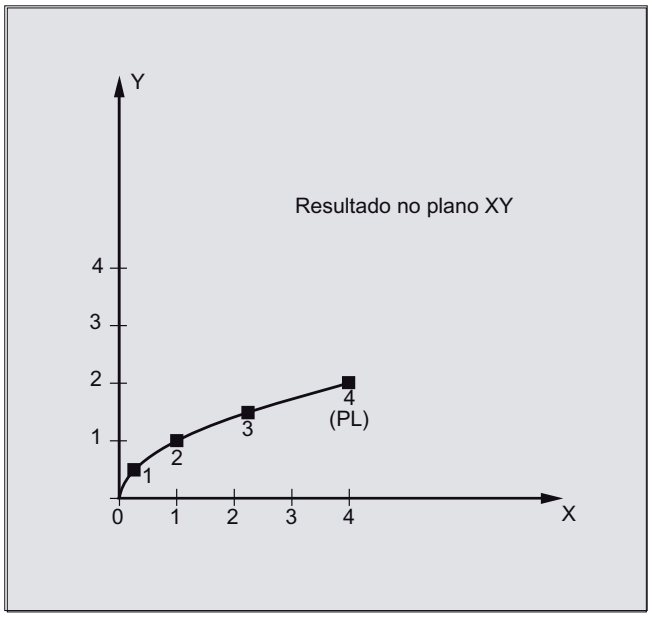
Programação

```
Código de programa
N9 X0 Y0 G90 F100
N10 POLY PO[Y]=(2) PO[X]=(4,0.25) PL=4
```

Trajetória das curvas X(p) e Y(p)



Trajetória da curva no plano XY



Descrição

A equação geral da função de polinômio é:

$f(p)= a_0 + a_1p + a_2p^2 +. . . + a_np^n$

com a_n: coeficientes constantes
: p: Parâmetros

No comando numérico podem ser programados no máx. polinômios de 5º grau:

$f(p)= a_0 + a_1p + a_2p^2 + a_3p^3 + a_4p^4 + a_5p^5$

Ao atribuir os coeficientes com valores concretos, podem ser criadas as mais diferentes formas de curvas, como retas, parábolas e funções exponenciais.

Uma reta é gerada através de $a_2 = a_3 = a_4 = a_5 = 0$:

$f(p) = a_0 + a_1p$

Ainda se aplica:

a_0: Posição de eixo no fim do bloco precedente

$p = PL$

$a_1 = (x_E - a_0 - a_2*p^2 - a_3*p^3) / p$

É possível programar polinômios **sem** que a interpolação de polinômios seja ativada através do comando G `POLY`. Porém, neste caso os polinômios programados não serão interpolados, apenas os pontos finais programados dos eixos serão aproximados linearmente (`G1`). Somente depois de uma ativação explícita da interpolação de polinômios no programa de peça (`POLY`) é que os polinômios programados serão executados como tais.

Particularidade: Polinômio de denominador

Para eixos geométricos também se pode programar um polinômio de denominador comum sem especificar um nome de eixo com `PO[]=(…)`, isto é, se interpola o movimento dos eixos geométricos quociente de dois polinômios.

Com isso é possível representar de forma exata segmentos cônicos (circunferência, elipse, parábola, hipérbole).

Exemplo:

Código de programa	Comentário
`POLY G90 X10 Y0 F100`	`; Eixos geométricos movimentam-se linearmente até a posição X10 Y0.`
`PO[X]=(0,-10) PO[Y]=(10) PO[]=(2,1)`	`; Eixos geométricos movimentam-se em quadrante até X0 Y10.`

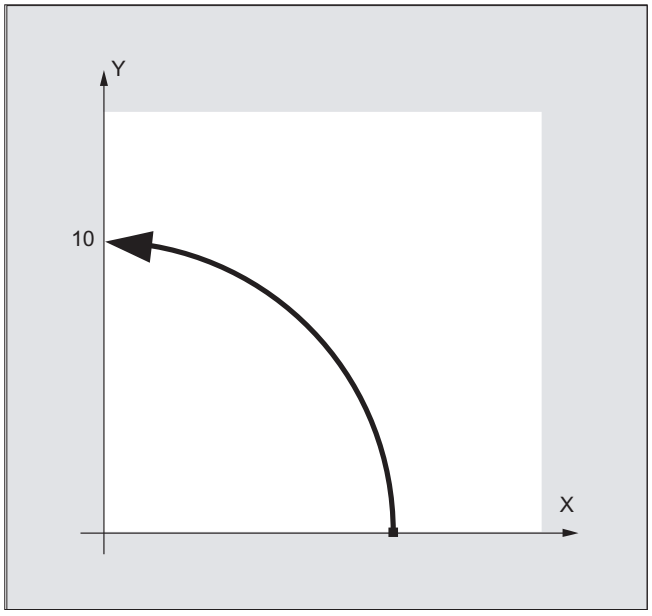
O coeficiente constante (a_0) do polinômio de denominador sempre é adotado com 1. O ponto final programado independe do `G90` / `G91`.

A partir dos valores programados o X(p) e U(p) são calculados como:

$X(p) = (10 - 10 * p^2) / (1 + p^2)$
$Y(p) = 20 * p / (1 + p^2)$
com $0 \leq p \leq 1$

Com base na programação dos pontos iniciais, pontos finais, coeficiente a_2 e PL=1 temos os seguintes resultados intermediários:

Numerador (X) = $10 + 0 * p - 10 * p^2$
Numerador (Y) = $0 + 20 * p + 0 * p^2$
Denominador = $1 + p^2$



É emitido um alarme se for programado um polinômio de denominador com zeros dentro do intervalo `[0,PL]`. O polinômio de denominador não tem nenhum efeito sobre o movimento de eixos adicionais.

Indicação

Uma correção do raio de compensação pode ser ativada com `G41` e `G42` em conjunto com a interpolação de polinômios, depois ser aplicada para interpolação linear ou circular.

4.6 Referência ajustável do percurso (SPATH, UPATH)

Função

Durante a interpolação o usuário pode fazer duas diferentes relações entre os eixos FGROUP determinantes de velocidade e os demais eixos de percurso: Por último, devem ser conduzidos ou sincronizados com o percurso S ou sincronizados com o parâmetro de curva U dos eixos FGROUP.

Ambos os tipos de interpolação de percurso são usados em diferentes aplicações e podem ser comutados entre eles através dos dois comandos de linguagem ativados modalmente no 45º grupo de códigos G, o `SPATH` e o `UPATH`.

Sintaxe

```
SPATH
UPATH
```

Significado

`SPATH`:	A referência de percurso para eixos FGROUP é o comprimento do arco
`UPATH`:	A referência de percurso para eixos FGROUP é o parâmetro de curva

Indicação

`UPATH` e `SPATH` também determinam a relação do polinômio de palavra F (`FPOLY`, `FCUB`, `FLIN`) com o movimento de percurso.

Condições gerais

A referência de percurso ajustada não tem significado:

- na interpolação linear e circular
- em blocos de rosca
- quando todos os eixos de percurso estão contidos no FGROUP.

Exemplos

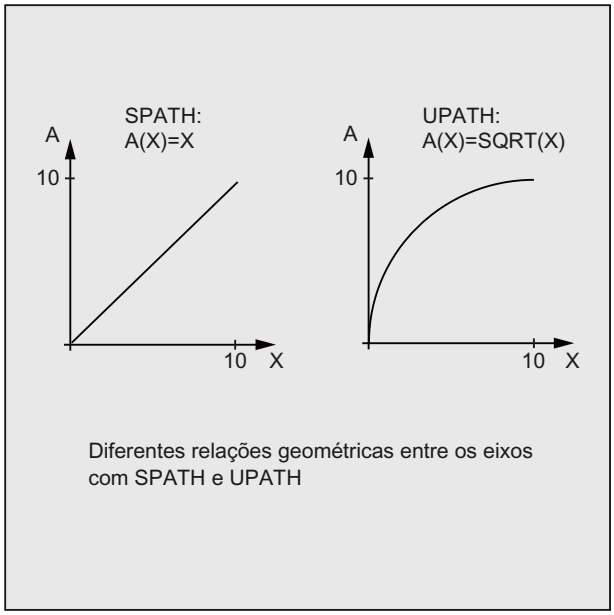
Exemplo 1:

No exemplo a seguir em um quadrado com 20 mm de lado se realiza a suavização com G643. Os desvios máximos do contorno exato são definidos para cada eixo através do dado de máquina específico de eixo MD33100 $MA_COMPRESS_POS_TOL[<n>].

Código de programa	**Comentário**
N10 G1 X… Y… Z… F500	
N20 G643	; Suavização interna de bloco com G643
N30 XO Y0	
N40 X20 Y0	; Comprimento de canto (mm) para os eixos
N50 X20 Y20	
N60 X0 Y20	
N70 X0 Y0	
N100 M30	

Exemplo 2:

O exemplo a seguir ilustra a diferença entre os dois tipos de controle de movimento. Nas duas vezes o pré-ajuste FGROUP(X,Y,Z) está ativo.



Diferentes relações geométricas entre os eixos com SPATH e UPATH

Código de programa

```
N10 G1 X0 A0 F1000 SPATH
N20 POLY PO[X]=(10,10) A10
```

Ou seja:

Código de programa

```
N10 G1 X0 F1000 UPATH
N20 POLY PO[X]=(10,10) A10
```

No bloco N20 o curso S dos eixos FGROUP depende do quadrado do parâmetro de curva U. Por isso que ao longo do curso de X resultam diferentes posições do eixo sincronizado A, dependendo de quem estiver ativo, SPATH ou UPATH.

Outras informações

Durante a interpolação de polinômios - e aqui entendemos a interpolação de polinômios no seu sentido direto (POLY), todos os tipos de interpolação Spline (ASPLINE, BSPLINE, CSPLINE) e interpolação linear com função de compressor (COMPON, COMPCURV) - as posições de todos os eixos de percurso i são determinadas através de polinômios pi(U). Neste caso, o parâmetro de curva U se move de 0 até 1 dentro de um bloco NC, sendo, portanto, normalizado.

Através do comando de linguagem FGROUP, dentro dos eixos de trajetória, podem ser selecionados os eixos que devem relacionar-se com o avanço programado. Porém, uma interpolação com velocidade constante no curso S destes eixos normalmente significa uma modificação não constante do parâmetros de curva U durante a interpolação de polinômios.

Comportamento de comando com Reset e dados de máquina e dados de opcionais

Após o Reset o código G (45º grupo de códigos G) determinado pelo MD20150 $MC_GCODE_RESET_VALUES[44] está ativo. Para permanecer compatível com instalações existentes, SPATH é ajustado como valor padrão.

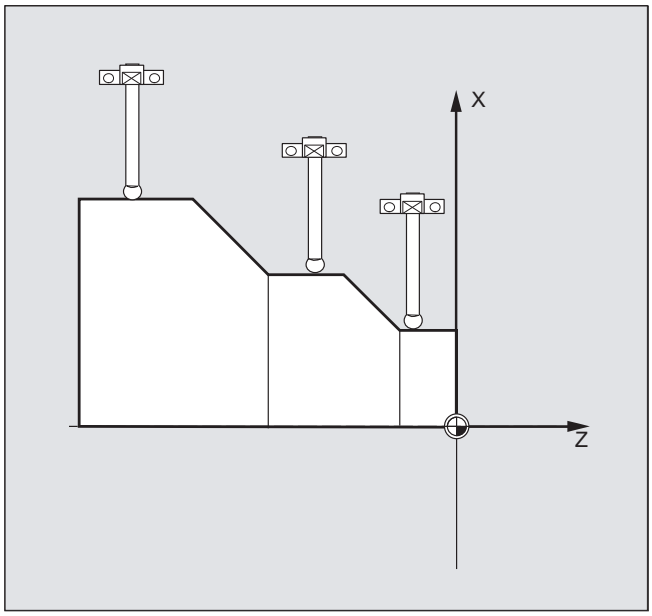
O valor de ajuste básico para o tipo de suavização é definido com o MD20150 $MC_GCODE_RESET_VALUES[9] (10º grupo de códigos G).

O dado de máquina específico de eixo MD33100 $MA_COMPRESS_POS_TOL[<n>] não possui um significado maior: ele contém as tolerâncias para a função de compressão e para a suavização com o G642.

4.7 Medição com apalpador comutável (MEAS, MEAW)

Função

Com a função "Medição com apalpador comutável" as posições reais na peça de trabalho são aproximadas e na borda de contato do apalpador de medição é realizada a medição das posições para todos os eixos programados no bloco de medição e depois realizada a gravação dos valores para cada eixo na célula de memória correspondente.



Programação de blocos de medição

Para a programação da função estão disponíveis os seguintes dois comandos:

- `MEAS`

  Com o comando `MEAS` é cancelado o curso restante entre a posição real e a posição nominal.

- `MEAW`

  O comando `MEAW` é empregado em tarefas de medição onde a posição programada deve ser aproximada em todas as ocasiões.

O `MEAS` e o `MEAW` estão ativos por blocos e são programados juntos com instruções de movimento. Os avanços e tipos de interpolação (G0, G1, ...), assim como o número de eixos, devem ser adaptados ao respectivo problema de medição.

Leitura dos resultados de medição

Os resultados de medição para os eixos analisados com apalpador de medição estão à disposição nas seguintes variáveis:

- `$AA_MM[<eixo>]`
  Resultados de medição no sistema de coordenadas da máquina
- `$AA_MW[<eixo>]`
  Resultados de medição no sistema de coordenadas da peça de trabalho

Não se gera nenhuma parada de pré-processamento durante a leitura destas variáveis.

Indicação

Com `STOPRE` deve ser programada uma parada de pré-processamento na posição adequada do programa NC. Caso contrário podem ser lidos valores incorretos.

Sintaxe

```
MEAS=<TE> G... X... Y... Z...
MEAW=<TE> G... X... Y... Z...
```

Significado

`MEAS` Comando: Medição **com** anulação do curso restante
Efeito: por blocos

`MEAW` Comando: Medição **sem** anulação do curso restante
Efeito: por blocos

`<TE>` Evento de disparo para partida da medição
Tipo: INT
Faixa de valores: -2, -1, 1, 2

Nota:
Existem no máximo 2 apalpadores de medição (dependendo do nível de expansão).

Significado:

(+)1	Flancos crescentes do apalpador de medição 1 (na entrada de medição 1)
-1	Flancos decrescentes do apalpador de medição 1 (na entrada de medição 1)
(+)2	Flancos crescentes do apalpador de medição 2 (na entrada de medição 2)
-2	Flancos decrescentes do apalpador de medição 2 (na entrada de medição 2)

Nota:
Existem no máximo 2 apalpadores de medição (dependendo do nível de expansão).

`G...` Tipo de interpolação, p. ex. `G0`, `G1`, `G2` ou `G3`

`X... Y... Z...` Ponto final em coordenadas cartesianas

Exemplo

Código de programa	Comentário
N10 MEAS=1 G1 F1000 X100 Y730 Z40	; Bloco de medição com apalpador de medição da primeira entrada de medição e interpolação de retas. A parada de pré-processamento é gerada automaticamente.
...	

Outras informações

Estado do pedido de medição

Se no programa for necessária uma avaliação para saber se o apalpador de medição realmente está ativo, então se pode consultar a variável de estado `$AC_MEA[n]` (n= número do apalpador):

Valor	Significado
0	Tarefa de medição não preenchida
1	Tarefa de medição concluída com sucesso (o apalpador de medição foi acionado)

Indicação

Quando o apalpador de medição for defletido, no programa, a variável passa para 1. Na partida de um bloco de medição a variável é automaticamente passada para o estado inicial do apalpador.

Coleta de valores de medição

São coletadas as posições de todos eixos de percurso e de posicionamento do bloco (número máximo de eixos está em função da configuração do comando). Com o `MEAS` o movimento é desacelerado controladamente depois de ser ativado o apalpador de medição.

Indicação

Se em um bloco de medição foi programado um eixo geométrico, então são armazenados os valores de medição dos atuais eixos geométricos.

Se em um bloco de medição foi programado um eixo envolvido em uma transformação, então são armazenados os valores de medição de todos eixos envolvidos com esta transformação.

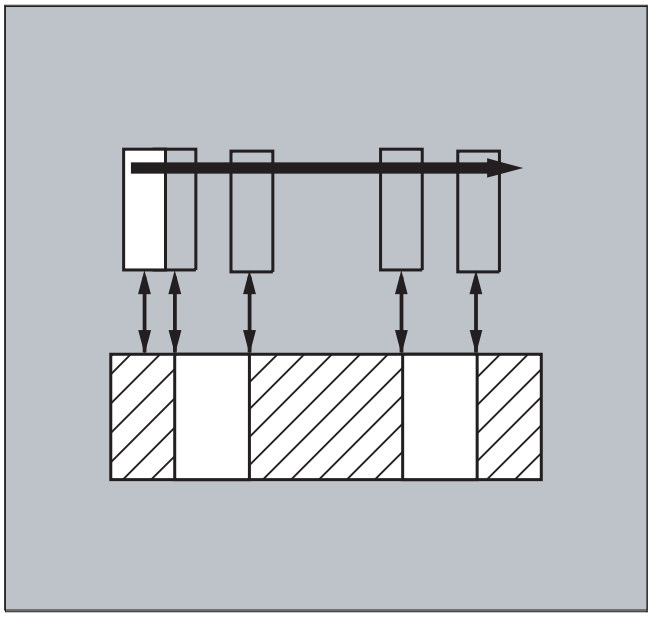
4.8 Função de medição ampliada (MEASA, MEAWA, MEAC) (opcional)

Função

Na medição por eixos podem ser usados vários apalpadores de medição e vários sistemas de medição.

Com o comando `MEASA` ou `MEAWA` são coletados até quatro valores de medição para o eixo programado em cada medição e estes são armazenados em variáveis de sistema conforme o evento de disparo.

Pedidos contínuos de medição podem ser realizados com o comando `MEAC`. Neste caso os resultados das medições são armazenados em variáveis FIFO. Também para `MEAC` são possíveis até quatro valores por medição.



Leitura dos resultados de medição

Os resultados de medição estão disponíveis nas seguintes variáveis:

- $AA_MM1...4[<eixo>]

  Resultados de medição no sistema de coordenadas da máquina

- $AA_MW1...4[<eixo>]

  Resultados de medição no sistema de coordenadas da peça de trabalho

Sintaxe

```
MEASA[<eixo>]=(<modo>,<TE1>,...,<TE4>)
MEAWA[<eixo>]=(<modo>,<TE1>,...,<TE4>)
MEAC[<eixo>]=(<modo>,<memória de medição>,<TE1>,...,<TE4>)
```

Indicação

MEASA e MEAWA estão ativos por blocos e podem ser programados juntos em um bloco. Se MEASA/MEAWA for programado junto com MEAS/MEAW em um bloco, será gerada uma mensagem de erro.

Significado

MEASA
Comando: Medição por eixo com anulação do curso restante
Efeito: por blocos

MEAWA
Comando: Medição por eixo sem anulação do curso restante
Efeito: por blocos

MEAC
Comando: Medição contínua por eixo sem anulação do curso restante
Efeito: por blocos

<eixo>
Nome do eixo de canal utilizado na medição

<modo>
Número de dois dígitos para especificação do modo de operação (modo de medição e sistema de medição)

Modo de medição (casa da unidade):

0 Cancelar pedido de medição.

1 Até 4 eventos de disparo diferentes ativados simultaneamente.

2 Até 4 eventos de disparo ativados sucessivamente.

3 Até 4 eventos de disparo ativados consecutivamente, mas sem monitoração do evento de disparo 1 no START (os alarmes 21700/21703 são suprimidos).

Nota:
Este modo não é possível com o MEAC.

Sistema de medição (casa da dezena):

0 (ou sem indicação) sistema de medição ativo

1 Sistema de medição 1

2 Sistema de medição 2

3 ambos sistemas de medição

<TE>
Evento de disparo para partida da medição

Tipo: INT

Faixa de valores: -2, -1, 1, 2

Significado:

(+)1 Flancos crescentes do apalpador de medição 1

-1 Flancos decrescentes do apalpador de medição 1

(+)2 Flancos crescentes do apalpador de medição 2

-2 Flancos decrescentes do apalpador de medição 2

<memória de medição>
Número do FIFO (memória de circulação)

Exemplos

Exemplo 1: Medição por eixo com anulação de curso restante em modo 1 (avaliação em ordem de tempo)

a) com 1 sistema de medição

```
Código de programa                          Comentário
...
N100 MEASA[X]=(1,1,-1) G01 X100 F100    ;  Medição no modo 1 com sistema de medição
                                           ativo. Esperar pelo sinal de medição com
                                           flanco crescente/decrescente do apalpador
                                           de medição 1 no percurso até X=100.
N110 STOPRE                             ;  Parada de pré-processamento
N120 IF $AC_MEA[1]==FALSE GOTOF FIM     ;  Controlar sucesso da medição.
N130 R10=$AA_MM1[X]                     ;  Armazenar valor de medição correspondente
                                           ao primeiro evento de disparo (flanco
                                           crescente) programado.
N140 R11=$AA_MM2[X]                     ;  Armazenar valor de medição correspondente
                                           ao segundo evento de disparo (flanco
                                           decrescente) programado.
N150 FIM:
```

b) com 2 sistemas de medição

```
Código de programa                          Comentário
...
N200 MEASA[X]=(31,1,-1) G01 X100 F100   ;  Medição no modo 1 com ambos sistemas de
                                           medição. Esperar pelo sinal de medição
                                           com flanco crescente/decrescente do
                                           apalpador de medição 1 no percurso até
                                           X=100.
N210 STOPRE                             ;  Parada de pré-processamento
N220 IF $AC_MEA[1]==FALSE GOTOF FIM     ;  Controlar sucesso da medição.
N230 R10=$AA_MM1[X]                     ;  Salvar o valor de medição do sistema de
                                           medição 1 com flanco crescente.
N240 R11=$AA_MM2[X]                     ;  Salvar o valor de medição do sistema de
                                           medição 2 com flanco crescente.
N250 R12=$AA_MM3[X]                     ;  Salvar o valor de medição do sistema de
                                           medição 1 com flanco de medição
                                           decrescente.
N260 R13=$AA_MM4[X]                     ;  Salvar o valor de medição do sistema de
                                           medição 2 com flanco de medição
                                           decrescente.
N270 FIM:
```

Exemplo 2: Medição por eixo com anulação de curso restante em modo 2 (avaliação em ordem de programação)

Código de programa	Comentário
...	
N100 MEASA[X]=(2,1,-1,2,-2) G01 X100 F100	; Medição no modo 2 com sistema de medição ativo. Esperar pelo sinal de medição na ordem dos flancos crescentes do apalpador de medição 1, flancos decrescentes do apalpador de medição 1, flancos crescentes do apalpador de medição 2, flancos decrescentes do apalpador de medição 2 no percurso até X=100.
N110 STOPRE	; Parada de pré-processamento
N120 IF $AC_MEA[1]==FALSE GOTOF APALPADOR2	; Controlar sucesso da medição com o apalpador de medição 1.
N130 R10=$AA_MM1[X]	; Armazenar valor de medição correspondente ao primeiro evento de disparo (flanco crescente do apalpador 1) programado.
N140 R11=$AA_MM2[X]	; Armazenar valor de medição correspondente ao segundo evento de disparo (flanco crescente do apalpador 1) programado.
N150 APALPADOR2:	
N160 IF $AC_MEA[2]==FALSE GOTOF FIM	; Controlar sucesso da medição com o apalpador de medição 2.
N170 R12=$AA_MM3[X]	; Armazenar valor de medição correspondente ao terceiro evento de disparo (flanco crescente do apalpador 2) programado.
N180 R13=$AA_MM4[X]	; Armazenar valor de medição correspondente ao quarto evento de disparo (flanco crescente do apalpador 2) programado.
N190 FIM:	

Exemplo 3: Medição contínua por eixo em modo 1 (avaliação em ordem de tempo)

a) Medição de até 100 valores medição

```
Código de programa                                  Comentário
...
N110 DEF REAL VALOR_MEDICAO[100]
N120 DEF INT loop=0
N130 MEAC[X]=(1,1,-1) G01 X1000 F100                ; Medição no modo 1 com sistema de medição ativo,
                                                      salvamento dos valores de medição sob $AC_FIFO1,
                                                      espera pelo sinal de medição com flancos
                                                      decrescentes do apalpador de medição 1 no percurso
                                                      até X=1000.
N135 STOPRE
N140 MEAC[X]=(0)                                    ; Cancelar a medição após alcançar a posição do eixo.
N150 R1=$AC_FIFO1[4]                                ; Armazenar número de valores de medição acumulados no
                                                      parâmetro R1.
N160 FOR loop=0 TO R1-1
N170 VALOR_MEDICAO[loop]=$AC_FIFO1[0]               ; Ler e armazenar valores de medição do $AC_FIFO1.
N180 ENDFOR
```

b) Medição com cancelamento de curso restante após 10 valores de medição

```
Código de programa                                                  Comentário
...
N10 WHEN $AC_FIFO1[4]>=10 DO MEAC[x]=(0) DELDTG(x)                  ; Anula o curso
                                                                      restante.
N20 MEAC[x]=(1,1,1,-1) G01 X100 F500
N30 MEAC[X]=(0)
N40 R1=$AC_FIFO1[4]                                                 ; Número de valores de
                                                                      medição.
...
```

Outras informações

Pedido de medição

A programação de uma tarefa de medição pode ser realizada no programa de peça ou a partir de uma ação síncrona (veja o capítulo "Ações síncronas de movimentos"). Apenas um pedido de medição pode estar ativo ao mesmo tempo para cada eixo.

Indicação

O avanço deve ser adaptado ao respectivo problema de medição.

Para `MEASA` e `MEAWA` os resultados corretos somente podem ser garantidos com avanços, nos quais não ocorre mais do que um mesmo evento de disparo e não mais do que 4 eventos diferentes de disparo em cada ciclo de controle de posição.

Na medição contínua com `MEAC` a relação entre o ciclo de interpolação e o ciclo de controle de posição não pode ser maior que 8 : 1.

Evento de disparo

Um evento de disparo é formado a partir do número do apalpador e o critério de ativação (flanco crescente ou decrescente) do sinal de medição.

Para cada medição podem ser processados até 4 eventos de disparo do apalpador de medição solicitado, portanto, até dois apalpadores de medição com dois flancos de medição cada. A seqüência do processamento assim como o número máximo de eventos de disparo depende do modo selecionado.

Indicação

Para modo de medição 1 aplica-se: Um mesmo evento de disparo somente pode ser programado uma vez em uma tarefa de medição!

Modo de operação

Com o primeiro dígito (casa da dezena) do modo de operação é selecionado o sistema de medição desejado. Se houver apenas um sistema de medição, mas houver um segundo programado, se aplica automaticamente o existente.

Com o segundo dígito (casa da unidade) é selecionado o modo de medição desejado. Com isso o processo de medição é adaptado às possibilidades do respectivo comando:

- **Modo 1**

  A avaliação dos eventos de disparo é realizada na seqüência cronológica da ocorrência. Neste modo apenas um evento de disparo pode ser programado para módulos de seis eixos, e no caso de haverem mais eventos de disparo especificados, passa-se automaticamente para o modo 2 (sem mensagem).

- **Modo 2**

  A avaliação dos eventos de disparo é realizada na seqüência programada.

- **Modo 3**

  A avaliação dos eventos de disparo é realizada na ordem programada, mas sem monitoração do evento de disparo 1 no START.

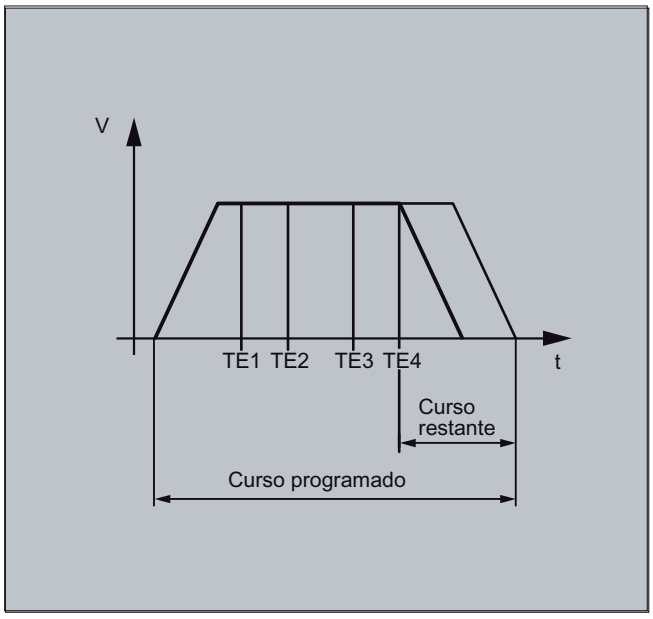
Indicação

No emprego de 2 sistemas de medição apenas se pode programar dois eventos de disparo.

Medição com e sem anulação de curso restante

Para a programação do MEASA a anulação de curso restante apenas será executada após a coleta dos valores de medição solicitados.

Para pedidos especiais de medição, onde para cada caso deve ser aproximada a posição programada, se aplica a função MEAWA.



Indicação

O MEASA não pode ser programado em ações síncronas. Como alternativa se pode programar o MEAWA mais a anulação de curso restante como ação síncrona.

Se o pedido de medição for iniciado com MEAWA a partir de ações síncronas, os valores de medição somente estarão disponíveis no sistema de coordenadas da máquina.

Resultados de medição para MEASA, MEAWA

Os resultados de medição estão disponíveis nas seguintes variáveis de sistema:

- No sistema de coordenadas da máquina:

`$AA_MM1[<eixo>]`	Valor de medição do sistema de medição programado com evento de disparo 1
`...`	...
`$AA_MM4[<eixo>]`	Valor de medição do sistema de medição programado com evento de disparo 4

- No sistema de coordenadas da peça de trabalho:

`$AA_WM1[<eixo>]`	Valor de medição do sistema de medição programado com evento de disparo 1
`...`	...
`$AA_WM4[<eixo>]`	Valor de medição do sistema de medição programado com evento de disparo 4

Indicação

Não se gera nenhuma parada de pré-processamento durante a leitura destas variáveis. Com `STOPRE` deve ser programada uma parada de pré-processamento na posição adequada. Caso contrário são lidos valores incorretos.

Eixos geométricos / Transformações

Se a medição axial deve ser iniciada para um eixo geométrico, deve ser programada explicitamente o mesmo pedido para os demais eixos geométricos. O mesmo vale para os eixos envolvidos em uma transformação.

Exemplo:

```
N10 MEASA[Z]=(1,1) MEASA[Y]=(1,1) MEASA[X]=(1,1) G0 Z100
```

ou

```
N10 MEASA[Z]=(1,1) POS[Z]=100
```

Pedido de medição com 2 sistemas de medição

Se um pedido de medição for executado com dois sistemas de medição, então se coleta os dois possíveis eventos de disparo de ambos sistemas de medição do respectivo eixo. Com isso se especifica a atribuição das variáveis reservadas:

`$AA_MM1[<eixo>]`	ou	`$AA_MW1[<eixo>]`	Valor de medição do sistema de medição no evento de disparo 1
`$AA_MM2[<eixo>]`	ou	`$AA_MW2[<eixo>]`	Valor de medição do sistema de medição no evento de disparo 2
`$AA_MM3[<eixo>]`	ou	`$AA_MW3[<eixo>]`	Valor de medição do sistema de medição 1 no evento de disparo 2
`$AA_MM4[<eixo>]`	ou	`$AA_MW4[<eixo>]`	Valor de medição do sistema de medição 2 no evento de disparo 2

Estado do apalpador de medição

O estado de medição está disponível nas seguintes variáveis de sistema:

$A_PROBE[<n>]

<n>=apalpador de medição

Valor	Significado
1	Apalpador de medição defletido
0	Apalpador de medição não defletido

Estado do pedido de medição no MEASA, MEAWA

Quando uma avaliação for necessária, então no programa pode ser consultado o estado do pedido de medição através do $AC_MEA[<n>], com <n> = número do apalpador de medição. Assim que todos os eventos de disparo do apalpador de medição <n> programados em um bloco forem executados, esta variável retorna o valor 1. Caso contrário, retorna-se o valor 0.

Indicação

O $AC_MEA não será mais atualizado se a medição for iniciada a partir de ações síncronas. Neste caso devem ser consultados novos sinais de estado PLC DB31, … DBX62.3 ou a variável de mesmo valor $AA_MEAACT[<eixo>].

Significado:

$AA_MEAACT==1: Medição ativa

$AA_MEAACT==0: Medição não ativa

Medição contínua (MEAC)

Os valores de medição estão disponíveis no MEAC no sistema de coordenadas da máquina e são armazenados na memória FIFO[n] (memória de circulação) especificada. Se forem configurados dois apalpadores para realizar a medição, os valores de medição do segundo apalpador são armazenados separadamente na memória FIFO[n] especialmente configurada (através de MD).

A memória FIFO é uma memória de circulação na qual os valores medidos são escritos em variáveis $AC_FIFO de acordo com o princípio de circulação, veja o capítulo "Ações síncronas de movimentos".

Indicação

O conteúdo FIFO somente pode ser lido uma vez da memória de circulação. Para a aplicação múltipla dos dados de medição, estes deverão ser armazenados temporariamente nos dados do usuário.

Se o número de valores de medição exceder o número máximo definido no dado da máquina para a memória FIFO, então a medição é encerrada automaticamente.

A medição contínua (sem fim) pode ser realizada através da leitura cíclica dos valores de medição. Neste caso a leitura de dados deve ser realizada na mesma freqüência da entrada dos novos valores de medição.

Programações detectadas com erro

As seguintes programações com erro são detectadas e exibidas com um erro:

- MEASA/MEAWA programado junto com MEAS/MEAW em um bloco

  Exemplo:

  ```
  N01 MEAS=1 MEASA[X]=(1,1) G01 F100 POS[X]=100
  ```

- MEASA/MEAWA com número de parâmetros <2 ou >5

  Exemplo:

  ```
  N01 MEAWA[X]=(1) G01 F100 POS[X]=100
  ```

- MEASA/MEAWA com evento de disparo diferente de 1/ -1/ 2/ -2

  Exemplo:

  ```
  N01 MEASA[B]=(1,1,3) B100
  ```

- MEASA/MEAWA em modo errado

  Exemplo:

  ```
  N01 MEAWA[B]=(4,1) B100
  ```

- MEASA/MEAWA com evento de disparo programado duas vezes

  Exemplo:

  ```
  N01 MEASA[B]=(1,1,-1,2,-1) B100
  ```

- MEASA/MEAWA e faltando eixo geométrico (GEO)

  Exemplo:

  ```
  N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) G01 X50 Y50 Z50 F100 ;Eixos GEO X/Y/Z
  ```

- Pedido de medição inconsistente com eixos geométricos

  Exemplo:

  ```
  N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) MEASA[Z]=(1,1,2) G01 X50 Y50 Z50 F100
  ```

4.9 Funções especiais para usuários OEM (OEMIPO1, OEMIPO2, G810 até G829)

Função

Endereços OEM

O significado dos endereços OEM é definida pelo usuário OEM. A funcionalidade é incorporada através de ciclos de compilação. Estão reservados 5 endereços OEM. O identificador de endereço pode ser ajustado. Endereços OEM são permitidos em cada bloco.

Parâmetros

Grupos G reservados

Grupo 1 com `OEMIPO1`, `OEMIPO2`

O usuário OEM pode definir dois nomes adicionais de funções G, o `OEMIPO1` e o `OEMIPO2`. Esta funcionalidade é incorporada através de ciclos de compilação e está reservada para o usuário OEM.

- Grupo 31 com **G810 até G819**
- Grupo 32 com **G820 até G829**

Para o usuário OEM estão reservados dois grupos G com 10 funções G OEM cada. **Com isso as funções incorporadas pelo usuário OEM podem ser acessadas para aplicações externas.**

Funções e subrotinas

Além disso os usuários OEM também podem criar funções e subrotinas pré-definidas com transferência de parâmetros.

4.10 Redução de avanço com desaceleração nos cantos (FENDNORM, G62, G621)

Função

Na desaceleração automática nos cantos o avanço é breve e gradativamente reduzido antes de alcançar o respectivo canto. Além disso se pode parametrizar a dimensão do comportamento de ferramenta relevante à usinagem através de dados de ajuste. São eles:

- Início e fim da redução do avanço
- Override com o qual o avanço é reduzido
- Detecção do canto relevante

Como cantos relevantes consideramos os cantos cujo ângulo interno é menor do que o canto projetado através do dado de ajuste.

A função do override automático de cantos é desativada com o valor padrão `FENDNORM`.

Literatura:
/FBFA/ Descrição do funcionamento dos dialetos ISO

Sintaxe

```
FENDNORM
G62 G41
G621
```

Significado

`FENDNORM`	Desaceleração automática de cantos desativada
`G62`	Desaceleração nos cantos internos com correção de raio da ferramenta ativada
`G621`	Desaceleração em todos cantos com correção de raio da ferramenta ativada

G62 apenas tem efeito nos cantos internos com

- correção de raio de ferramenta ativada `G41`, `G42` e
- modo de controle da trajetória ativado `G64`, `G641`

O respectivo canto é aproximado com avanço reduzido, resultante de:

F * (override para redução do avanço) * override de avanço

A redução de avanço máxima possível é alcançada exatamente quando a ferramenta realizar a mudança de sentido no respectivo canto, com referência ao centro do percurso.

G621 age de forma semelhante ao G62 em cada canto dos eixos definidos por `FGROUP`.

4.11 Critérios programáveis para fim de movimentos (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA)

Função

Semelhante ao critério de mudança de blocos na interpolação de percurso (G601, G602 e G603), com a interpolação individual de eixo, o critério de fim de movimento pode ser programado em um programa de peças ou ações síncronas para eixos de comando e PLC.

Dependendo de qual critério de fim de movimento for programado, os blocos do programa de peça e os blocos de ciclo de tecnologia com movimentos individuais de eixo requerem tempos diferentes para serem concluídos. O mesmo se aplica para o PLC através do FC15/ 16/ 18.

Sintaxe

```
FINEA[<eixo>]
COARSEA[<eixo>]
IPOENDA[<eixo>]
IPOBRKA(<eixo>[,<momento>])
ADISPOSA(<eixo>[,<modo>,<tamanho de janela>])
```

Significado

`FINEA`:	Fim de movimento ao alcançar "Parada exata fina"
`COARSEA`:	Fim de movimento ao alcançar "Parada exata aproximada"
`IPOENDA`:	Fim de movimento ao alcançar "Parada de interpolador"
`IPOBRKA`:	Mudança de blocos possível na rampa de frenagem
`ADISPOSA`:	Tamanho da janela de tolerância para critério de fim de movimento
`<eixo>`:	Nome de eixo de canal (X, Y, ...)
`<momento>`:	Momento da mudança de blocos relacionado à rampa de frenagem em %
`<modo>`:	Modo Tipo: INT Faixa de valores: 0 Janela de tolerância não ativa 1 Janela de tolerância em relação à posição nominal 2 Janela de tolerância em relação à posição real
`<tamanho de janela>`:	Tamanho da janela de tolerância Este valor é registrado no dado de ajuste SD43610 $SA_ADISPOSA_VALUE de forma sincronizada ao processamento principal. Tipo: REAL

Exemplos

Exemplo 1: Fim de movimento ao alcançar a parada de interpolador

```
Código de programa                                                    Comentário
...
N110 G01 POS[X]=100 FA[X]=1000 ACC[X]=90 IPOENDA[X]                   Movimento até a
                                                                      posição X100 com uma
                                                                      velocidade de
                                                                      percurso de 1000 rpm
                                                                      com um valor de
                                                                      aceleração de 90% e
                                                                      o fim de movimento
                                                                      ao alcançar a parada
                                                                      de interpolador.
...
N120 EVERY $A_IN[1] DO POS[X]=50 FA[X]=2000 ACC[X]=140 IPOENDA[X]     ; Movimento até a
                                                                      posição X50 quando a
                                                                      entrada 1 estiver
                                                                      ativa, com uma
                                                                      velocidade de
                                                                      percurso de 2000 rpm
                                                                      e um valor de
                                                                      aceleração de 140% e
                                                                      o fim de movimento
                                                                      ao alcançar a parada
                                                                      de interpolador.
...
```

Exemplo 2: Critério de mudança de blocos da rampa de frenagem no programa de peça

```
Código de programa          Comentário
                            ; Ajuste padrão ativo
N40 POS[X]=100              ; A mudança de bloco é realizada quando o eixo X alcançar a posição 100 e
                              parada exata fina.
N20 IPOBRKA(X,100)          ; Ativar critério de mudança de blocos da rampa de frenagem.
N30 POS[X]=200              ; A mudança de blocos ocorre assim que o eixo X começar a frear.
N40 POS[X]=250              ; O eixo X não freia na posição 200, mas ele continua se movendo até a
                              posição 250, assim que o eixo X começar a desacelerar é iniciada a
                              mudança de blocos.
N50 POS[X]=0                ; O eixo X freia e retorna para a posição 0 e a mudança de blocos ocorre
                              na posição 0 e parada exata fina.
N60 X10 F100
N70 M30
...
```

Exemplo 3: Critério de mudança de blocos da rampa de frenagem em ações síncronas

```
Código de programa          Comentário
                            ; No ciclo de tecnologia:
FINEA                       ; Critério de fim de movimento da parada exata fina.
POS[X]=100                  ; A mudança de blocos do ciclo de tecnologia ocorre quando o eixo X
                              alcançar a posição 100 e parada exata fina.
IPOBRKA(X,100)              ; Ativar critério de mudança de blocos da rampa de frenagem.
POS[X]=100                  ; POS[X]=100; a mudança de blocos do ciclo de tecnologia ocorre assim que
                              o eixo X começar a frear.
POS[X]=250                  ; O eixo X não freia na posição 200, mas continua o movimento até a
                              posição 250, assim que o eixo X começar a desacelerar ocorre a mudança
                              de blocos no ciclo de tecnologia.
POS[X]=250                  ; O eixo X freia e retorna para a posição 0 e a mudança de blocos ocorre
                              na posição 0 e parada exata fina.
M17
```

Outras informações

Leitura do critério de fim de movimento

O critério de fim de movimento pode ser consultado com a variável$AA_MOTEND[<eixo>].

Valor	Significado
1	Fim de movimento com "Parada exata fina"
2	Fim de movimento com "Parada exata aproximada"
3	Fim de movimento com "Parada de IPO".
4	Critério de mudança de blocos, rampa de frenagem do movimento do eixo
5	Mudança de blocos na rampa de frenagem com janela de tolerância em relação à "Posição nominal"
6	Mudança de blocos na rampa de frenagem com janela de tolerância em relação à "Posição real"

Indicação

O último valor programado é preservado após RESET.

Critério de mudança de blocos na rampa de frenagem

O valor porcentual é registrado no dado de ajuste de modo síncrono no processamento principal:

SD43600 $SA_IPOBRAKE_BLOCK_EXCHANGE

Se não for especificado nenhum valor, será ativado o atual valor deste dado de ajuste.

Se pode ajustar em uma faixa de 0 % até 100 %.

Janela adicional de tolerância para IPOBRKA

Para o critério de mudança de blocos existente na rampa de frenagem também pode ser selecionada uma "janela de tolerância" adicional de critério de mudança de blocos. A habilitação somente ocorre quando o eixo:

- como de costume alcançar o valor % de sua rampa de frenagem

  e

- tiver sua atual posição real ou nominal não mais afastada do que uma tolerância da posição final do eixo no bloco.

Literatura

Para mais informações sobre o critério de mudança de blocos dos eixos de posicionamento veja:

- Manual de funções ampliadas; Eixos de posicionamento (P2)
- Manual de programação Fundamentos; capítulo "Controle de avanço"

4.12 Bloco programável de parâmetros servo (SCPARA)

Função

Com SCPARA o bloco de parâmetros (composto por MDs) pode ser programado no programa de peça e em ações sincronizadas (antes possível apenas através do PLC).

DB3n DBB9 Bit3

Para evitar qualquer conflito entre o PLC e o NCK, se define mais um Bit na interface PLC → NCK:

DB3n DBB9 Bit3 "Especificação de bloco de parâmetros bloqueada pelo SCPARA".

Com uma especificação de bloco de parâmetros bloqueada para SCPARA não é emitida nenhuma mensagem de erro, caso esta ainda venha a ser programada.

Sintaxe

```
SCPARA[<eixo>]=<valor>
```

Significado

SCPARA	Definir bloco de parâmetros
<eixo>	Nome de eixo de canal (X, Y, ...)
<valor>	bloco de parâmetros desejado (1<= valor <=6)

Indicação

O atual bloco de parâmetros pode ser consultado na variável de sistema $AA_SCPAR[<eixo>].

Com G33, G331 ou G332 se seleciona o bloco de parâmetro mais adequado do comando.

Se o **bloco de parâmetro servo** deve ser mudado tanto em um programa de peça ou em uma ação sincronizada como no **PLC**, então se deve ampliar o programa PLC de usuário.

Literatura:
/FB1/ Manual de funções básicas; Avanços (V1),
Capítulo "Controle de avanço".

Exemplo

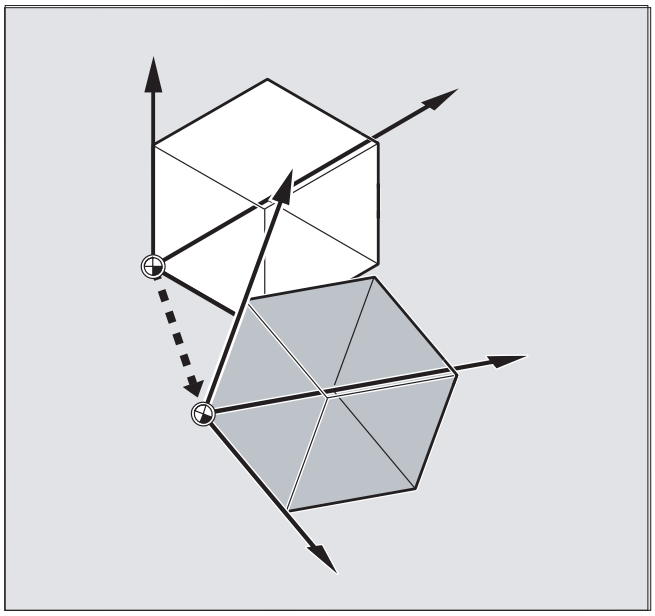
```
Código de programa     Comentário
...
N110 SCPARA[X]= 3      ; O 3° bloco de parâmetros é selecionado para o eixo X.
...
```

Transformações de coordenadas (FRAMES) 5

5.1 Transformação de coordenadas através de variável Frame

Função

Além das opções de programação descritas no Manual de programação "Fundamentos" também podemos definir sistemas de coordenadas com variáveis Frame pré-definidas.



Estão definidos os seguintes sistemas de coordenadas:

MCS: Sistema de coordenadas da máquina

BCS: Sistema de coordenadas básico

BNS: Sistema de coordenadas do ponto zero básico

ENS: Sistema ajustável de coordenadas do ponto zero

WCS: Sistema de coordenadas da peça

O que é uma variável Frame pré-definida?

Variáveis Frame pré-definidas são palavras-chave definidas na linguagem do comando com efeito correspondente, e que podem ser processadas no programa NC.

Possíveis variáveis Frame:

- Frame básico (deslocamento básico)
- Frames ajustáveis
- Frame programável

Ler atribuições de valores e valores reais

Relação variável Frame / Frame

Uma transformação de coordenadas pode ser ativada em uma variável Frame através da atribuição de valor de um Frame.

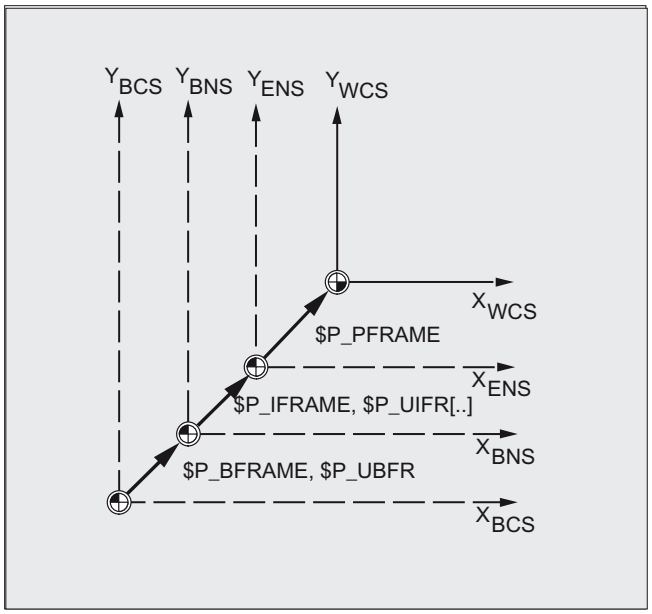
Exemplo: `$P_PFRAME=CTRANS(X,10)`

Variável Frame:

`$P_PFRAME` significa: atual Frame programável.

Frame:

`CTRANS(X,10)` significa: deslocamento de ponto zero programável do eixo X em 10 mm.



Leitura de valores reais

Os atuais valores reais do sistema de coordenadas podem ser lidos no programa de peça através de variáveis pré-definidas:

$AA_IM[eixo]: Leitura do valor real em MCS

$AA_IB[eixo]: Leitura do valor real em BCS

$AA_IBN[eixo:] Leitura do valor real em BNS

$AA_IEN[eixo]: Leitura do valor real em ENS

$AA_IW[eixo]: Leitura do valor real em WCS

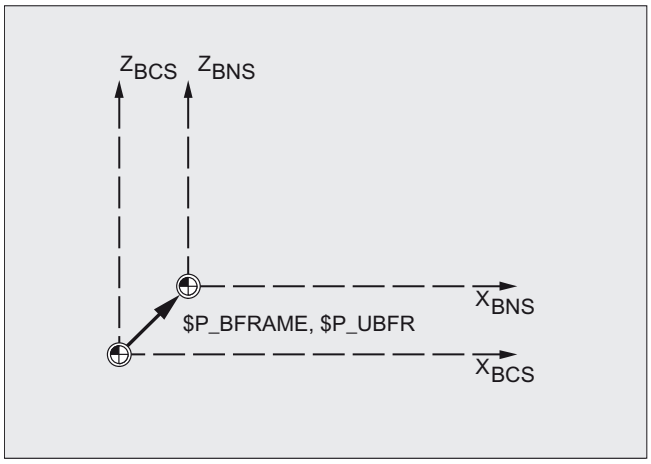
5.1.1 Variável Frame pré-definida ($P_BFRAME, $P_IFRAME, $P_PFRAME, $P_ACTFRAME)

$P_BFRAME

Atual variável Frame básica que estabelece a relação entre o sistema de coordenadas básico (BCS) e o sistema básico do ponto zero (BNS).

Se o Frame básico descrito pelo $P_UBFR deve ser ativado imediatamente no programa, ou se deve

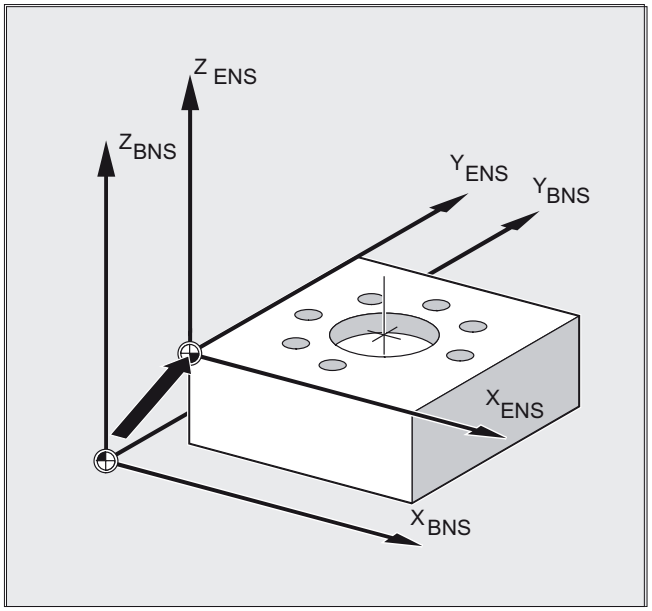
- programar um `G500`, `G54...G599` ou
- descrever um `$P_BFRAME` com `$P_UBFR`.



$P_IFRAME

Atual variável ajustável que estabelece a relação entre o sistema básico do ponto zero (BNS) e o sistema ajustável do ponto zero (ENS).

- `$P_IFRAME` corresponde ao `$P_UIFR[$P_IFRNUM]`
- `$P_IFRAME` depois da programação de p. ex. G54 contém a translação, rotação, escala e espelhamento definido por `G54`.

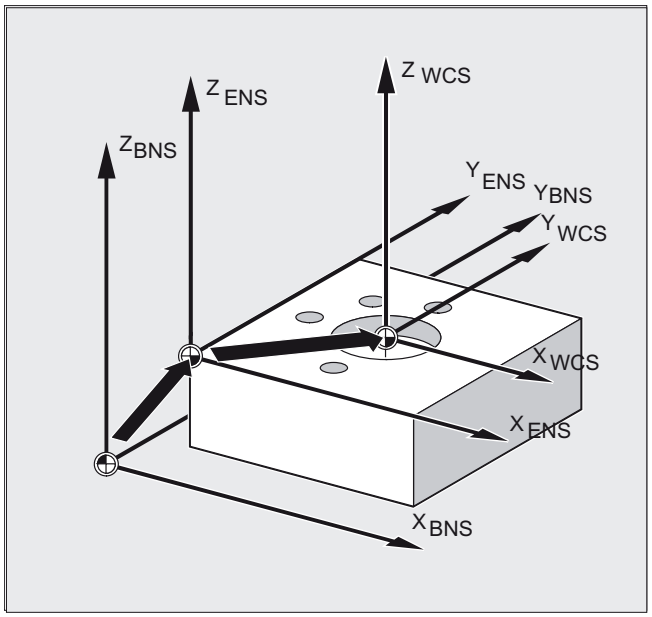


$P_PFRAME

Atual variável programável que estabelece a relação entre o sistema ajustável do ponto zero (ENS) e o sistema de coordenadas da peça de trabalho (WCS).

`$P_PFRAME` contém o Frame resultante, que

- resulta **da programação** do `TRANS/ATRANS`, `ROT/AROT`, `SCALE/ASCALE`, `MIRROR/AMIRROR` e
- **a partir da atribuição** do `CTRANS`, `CROT`, `CMIRROR`, `CSCALE` no FRAME programável



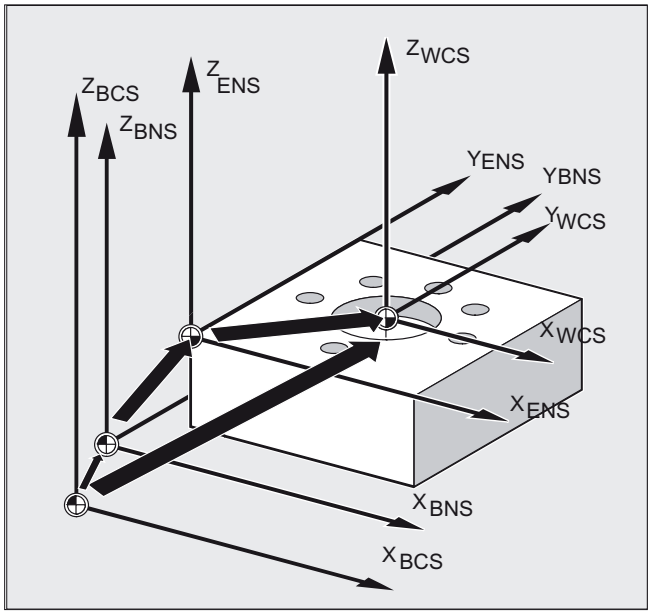
$P_ACTFRAME

Atual Frame total resultante, que resulta através do encadeamento

- da atual variável Frame básica `$P_BFRAME`,
- da atual variável Frame ajustável `$P_IFRAME` com Frames de sistema e
- da atual variável Frame programável `$P_PFRAME` com Frames de sistema

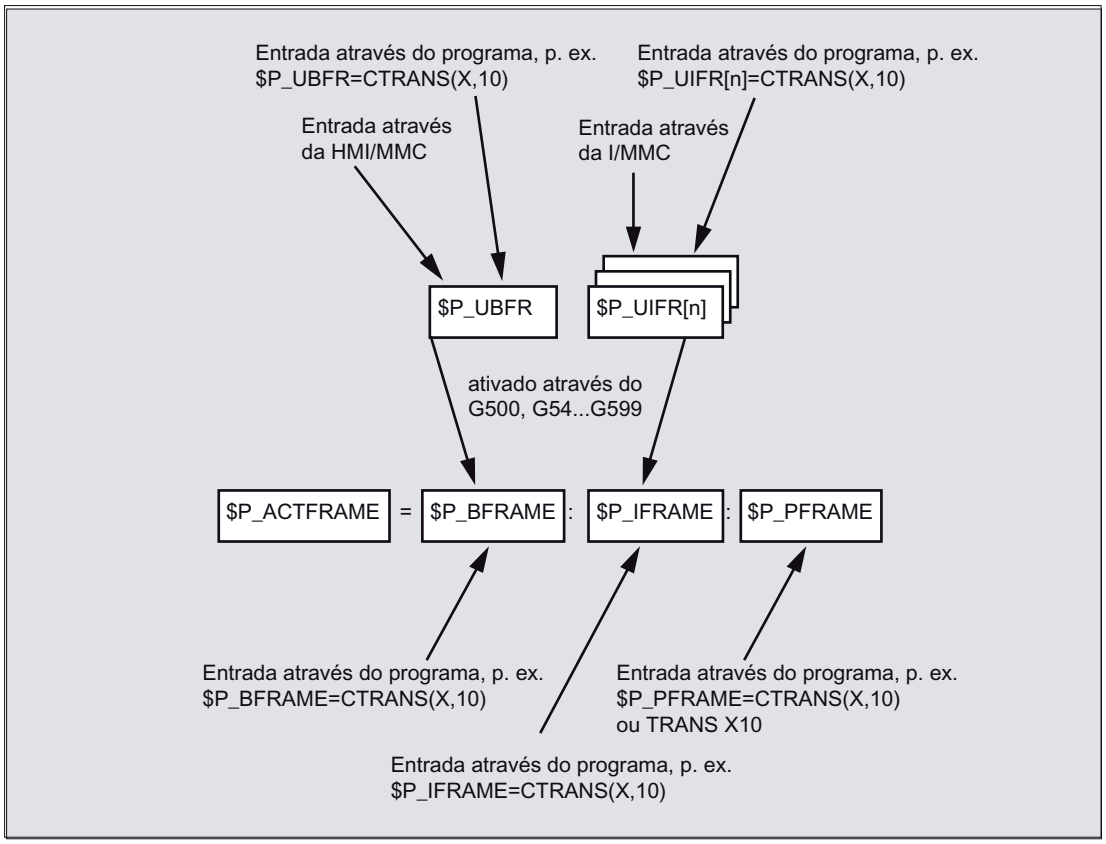
. Para Frames de sistema, veja o capítulo "Frames ativos no canal"

`$P_ACTFRAME` descreve o atual ponto zero válido da peça de trabalho.



Se `$P_BFRAME`, `$P_IFRAME` ou `$P_PFRAME` forem modificadas, o `$P_ACTFRAME` será recalculado.

`$P_ACTFRAME` corresponde ao `$P_BFRAME:$P_IFRAME:$P_PFRAME`



O Frame básico e o Frame ajustável tem efeito após Reset, se o MD 20110 RESET_MODE_MASK estiver ajustado da seguinte forma:

Bit0=1, Bit14=1 -->atua o `$P_UBFR` (Frame básico)

Bit0=1, Bit5=1 -->atua o `$P_UIFR[$P_UIFRNUM]` (Frame ajustável)

Frames ajustáveis pré-definidos $P_UBFR

Com $P_UBFR se programa o Frame básico, mas ele não está simultaneamente ativo no programa de peça. O Frame básico escrito com $P_UBFR é processado quando

- se ativa o Reset e os Bits 0 e 14 do MD RESET_MODE_MASK estiverem definidos,
- e foram executadas as instruções `G500`, `G54...G599`.

Frames ajustáveis pré-definidos $P_UIFR[n]

Através da variável Frame pré-definida $P_UIFR[n] os deslocamentos ajustáveis do ponto zero `G54` até `G599` podem ser lidos e gravados a partir do programa de peça.

Na estrutura estas variáveis representam um campo de uma dimensão de tipo FRAME com o nome `$P_UIFR[n]`.

Atribuição com os comandos G

Como padrão temos pré-definidos 5 Frames ajustáveis `$P_UIFR[0]... $P_UIFR[4]` e 5 comandos G de mesmo significado – `G500` e `G54` até `G57`, sendo que no endereço dos mesmos são armazenados os valores.

`$P_IFRAME=$P_UIFR[0]` corresponde ao `G500`

`$P_IFRAME=$P_UIFR[1]` corresponde ao `G54`

`$P_IFRAME=$P_UIFR[2]` corresponde ao `G55`

`$P_IFRAME=$P_UIFR[3]` corresponde ao `G56`

`$P_IFRAME=$P_UIFR[4]` corresponde ao `G57`

O número de Frames pode ser alterado através do dado de máquina:

`$P_IFRAME=$P_UIFR[5]` corresponde ao `G505`

`... ... ...`

`$P_IFRAME=$P_UIFR[99]` corresponde ao `G599`

Indicação

Com isso se pode criar ao todo 100 sistemas de coordenadas, que podem ser chamados para diversos dispositivos ao longo do programa.

⚠ CUIDADO

A programação das variáveis Frame e de Frames requer um bloco NC próprio no programa NC. **Exceção:** Programação de um Frame ajustável com `G54`, `G55`, `...`

5.2 Atribuir valores para variáveis Frame / Frame

5.2.1 Atribuir valores diretos (valor de eixo, ângulo, escala)

Função

No programa NC podemos atribuir valores diretamente em Frames ou em variáveis Frame.

Sintaxe

```
$P_PFRAME=CTRANS (X, valor de eixo, Y, valor de eixo, Z, valor de eixo, …)
$P_PFRAME=CROT (X, ângulo, Y, ângulo, Z, ângulo, …)
$P_UIFR[..]=CROT (X, ângulo, Y, ângulo, Z, ângulo, …)
$P_PFRAME=CSCALE (X, escala, Y, escala, Z, escala, …)
$P_PFRAME=CMIRROR (X, Y, Z)
```

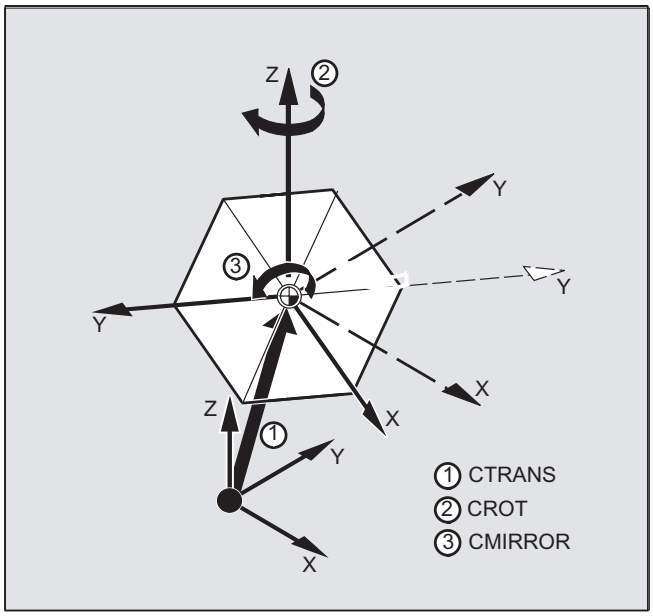
A programação do $P_BFRAME é feita da mesma forma como no $P_PFRAME.

Significado

`CTRANS`	Deslocamento nos eixos especificados
`CROT`	Rotação em torno dos eixos especificados
`CSCALE`	Alteração de escala nos eixos especificados
`CMIRROR`	Mudança de sentido dos eixos especificados
`X Y Z`	Valor de deslocamento no sentido do eixo geométrico especificado
`Valor de eixo`	Atribuir o valor de eixo do deslocamento
`Ângulo`	Atribuir o ângulo de rotação em torno dos eixos indicados
`Escala`	Alterar escala

Exemplo

Com a atribuição de valores no atual Frame programado se ativam as funções de translação, rotação e espelhamento.




```
N10 $P_PFRAME=CTRANS(X,10,Y,20,Z,5):CROT(Z,45):CMIRROR(Y)
```

Definir outros valores prévios para os componentes Frame-Rot.

Definir valores prévios com CROT para todos os três componentes do UIFR

Código de programa	Comentário
`$P_UIFR[5]=CROT (X, 0, Y, 0, Z, 0)`	
`N100 $P_UIFR[5, y, rt]=0`	
`N100 $P_UIFR[5, x, rt]=0`	
`N100 $P_UIFR[5, z, rt]=0`	

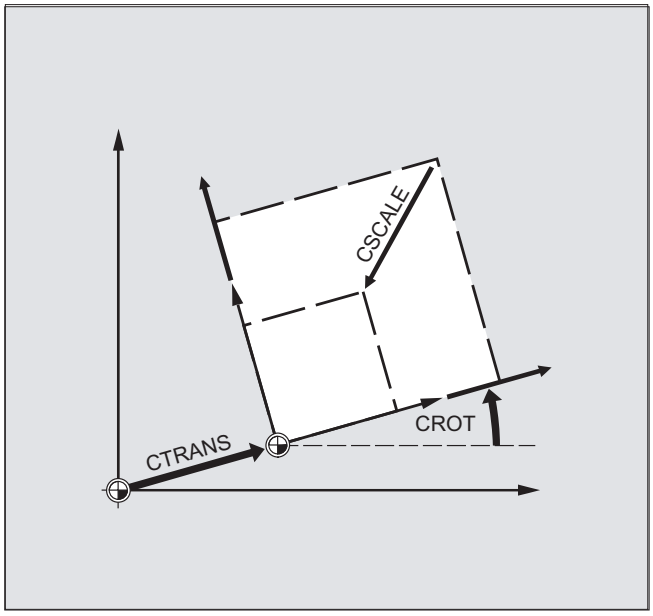
Descrição

Podemos programar várias diretrizes de cálculo consecutivamente.

Exemplo:

```
$P_PFRAME=CTRANS(...):CROT(...):CSCALE...
```

Observe que os comandos devem estar ligados entre si com o operador de concatenação, o duplo ponto (...):(...). Com isso os comandos são primeiramente ligados entre si e depois executados uma a um na seqüência programada.



Indicação

Os valores programados com os comandos mencionados são atribuídos aos Frames e armazenados.

Os valores somente serão ativados se eles forem atribuídos ao Frame de uma variável Frame `$P_BFRAME` ou `$P_PFRAME` ativa.

5.2.2 Ler e alterar componentes de Frame (TR, FI, RT, SC, MI)

Função

Temos a possibilidade de acessar **determinados** dados de um Frame como um determinado valor de deslocamento ou ângulo de rotação. Estes valores podem ser modificados ou atribuídos em outra variável.

Sintaxe

`R10=$P_UIFR[$P_UIFNUM,X,RT]`	O ângulo de rotação RT em torno do eixo X do atual deslocamento ajustável do ponto zero $P_UIFRNUM deve ser atribuído na variável R10.
`R12=$P_UIFR[25,Z,TR]`	O valor do deslocamento TR em Z do bloco de dados do Frame nº 25 ajustado deve ser atribuído na variável R12.
`R15=$P_PFRAME[Y,TR]`	O valor do deslocamento TR em Y do atual Frame programável deve ser atribuído na variável R15.
`$P_PFRAME[X,TR]=25`	O valor do deslocamento TR em X do atual Frame programável deve ser modificado. X25 é válido de imediato.

Significado

`$P_UIFRNUM`	`Com estas variáveis se estabelece automaticamente a relação do atual deslocamento ajustável de ponto zero válido.`
`P_UIFR[n,…,…]`	`Com a especificação do número de Frame n se pode acessar o Frame ajustável n° n.`
	`Indicação dos componentes que devem ser lidos ou modificados:`
`TR`	`TR Translation`
`FI`	`FI Translation Fine`
`RT`	`RT Rotation`
`SC`	`Alteração de escala SC Scale`
`MI`	`Espelhamento MI`
`X Y Z`	`Também (veja exemplos) é especificado o eixo X, Y e Z correspondente.`

Faixa de valores para rotação RT

Rotação em torno do 1º eixo geométrico:	-180° até +180°
Rotação em torno do 2º eixo geométrico:	-90° até +90°
Rotação em torno do 3º eixo geométrico:	-180° até +180°

Descrição

Chamar Frame

Com a indicação das variáveis $P_UIFRNUM pode-se acessar diretamente o atual deslocamento de ponto zero
(o $P_UIFRNUM contém o número do atual Frame definido) ajustado com $P_UIFR ou G54, G55, ...

Todos demais Frames ajustáveis $P_UIFR que estiverem armazenados serão chamados com a indicação do no respectivo número $P_UIFR[n].

Para as variáveis Frame pré-definidas e os Frames de definição própria especifique o nome, p. ex. $P_IFRAME.

Chamar dados

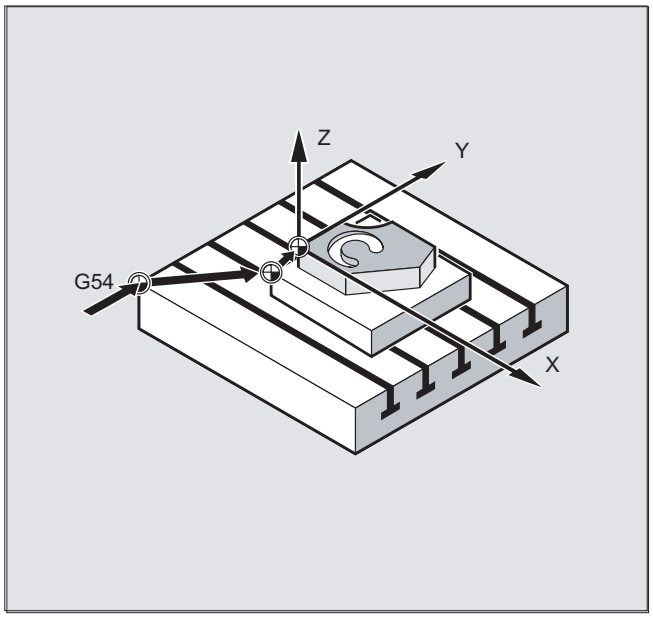
Entre os colchetes constam o nome do eixo e o componente de Frame do valor que se deseja acessar ou modificar, p. ex. [X, RT] ou [Z, MI].

5.2.3 Referências de Frames completos

Função

No programa NC pode ser atribuído um Frame completo à outro Frame ou Frames podem ser encadeados entre si.

Por exemplo, o encadeamento de Frames é indicado para descrição de várias peças de trabalho, que estão dispostas sobre um palete e devem ser usinadas em um processo de usinagem.



Para a descrição das tarefas dos paletes, por exemplo, os componentes de Frame poderiam conter apenas determinados valores parciais, através do encadeamento dos componentes podem ser gerados vários pontos zero da peça de trabalho.

Sintaxe

Atribuir Frames

```
DEF FRAME AJUSTE1                    ; Ao atual Frame programável são
AJUSTE1=CTRANS(X,10)                 ; atribuídos os valores do Frame
$P_PFRAME=AJUSTE1                    ; AJUSTE1 de definição própria.
DEF FRAME AJUSTE4                    ; O atual Frame programável é
AJUSTE4=$P_PFRAME                    ; armazenado na memória intermediária
$P_PFRAME=AJUSTE4                    ; e pode, se necessário, ser
                                     ; restaurado.
```

Encadeamento de Frames

Os Frames são encadeados entre si na seqüência programada, os componentes de Frame (deslocamentos, rotações, etc.) são executados aditivamente em sucessão.

$P_IFRAME=$P_UIFR[15]:$P_UIFR[16]	Por exemplo, o $P_UIFR[15] contém dados para deslocamentos de ponto zero. Em seguida são processados aditivamente os dados do $P_UIFR[16], como por exemplo os dados das rotações.
$P_UIFR[3]=$P_UIFR[4]:$P_UIFR[5]	O Frame ajustável 3 é gerado com o encadeamento dos Frames ajustáveis 4 e 5.

Indicação

Observe que os Frames precisam ser ligados entre si através do operador de encadeamento, o duplo ponto :.

5.2.4 Definição de novos Frames (DEF FRAME)

Função

Além dos Frames pré-definidos e ajustáveis descritos anteriormente, ainda existe a opção de se criar novos Frames. Neste caso tratam-se de variáveis de tipo FRAME que são definidas com qualquer nome.

Com as funções CTRANS, CROT, CSCALE e CMIRROR podemos atribuir nossos Frames com valores no programa NC.

Sintaxe

```
DEF FRAME PALETE1
PALETE1=CTRANS(…):CROT(…)…
```

Significado

`DEF FRAME`	Criar novos Frames.
`PALETE1`	Nome do novo Frame
`=CTRANS(...):` `CROT(...)...`	Atribuir valores às possíveis funções

5.3 Deslocamento aproximado e deslocamento fino (CFINE, CTRANS)

Função

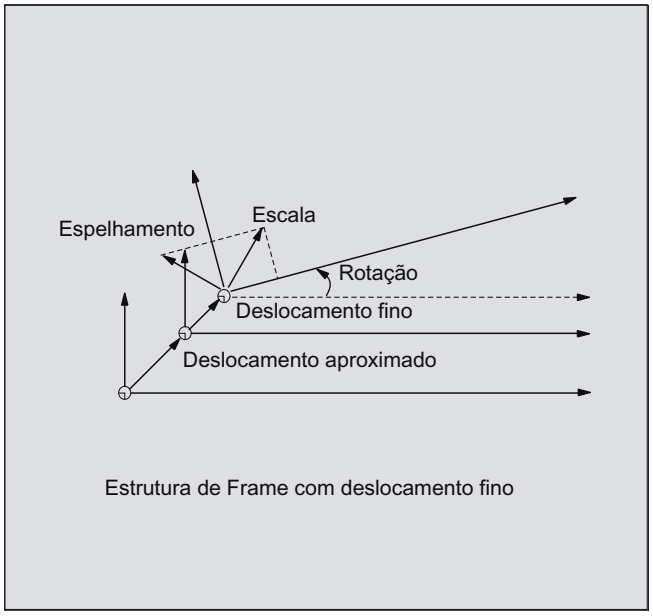
Deslocamento fino

Com o comando `CFINE (X, ..., Y, ...)` pode ser programado um deslocamento fino do Frame básico e todos Frames ajustáveis.

Um deslocamento fino somente pode ser realizado se o MD18600 $MN_MM_FRAME_FINE_TRANS=1.

Deslocamento aproximado

Com `CTRANS(...)` se define o deslocamento aproximado.



Estrutura de Frame com deslocamento fino

Os deslocamentos aproximado e fino são adicionados ao deslocamento total.

Sintaxe

```
$P_UBFR=CTRANS(x, 10) : CFINE(x, 0.1) :
CROT(x, 45)                                ;encadeamento do deslocamento,
                                           ;deslocamento fino e rotação
$P_UIFR[1]=CFINE(x, 0.5 y, 1.0, z, 0.1)    ;o Frame inteiro é sobrescrito com
                                           ;inclusive o deslocamento
                                           ;aproximado
```

O acesso aos componentes individuais do deslocamento fino é realizado pela especificação de componente FI (Translation Fine).

```
DEF REAL FINEX                          ;definição da variável FINEX
FINEX=$P_UIFR[$P_UIFNUM, x, FI]         ;leitura do deslocamento fino
                                        ;através da variável FINEX
FINEX=$P_UIFR[3, x, FI]$P               ;leitura do deslocamento fino
                                        ;do eixo X no 3º Frame
                                        ;através da variável FINEX
```

Significado

`CFINE(x, valor, y, valor, z, valor )`	Deslocamento fino para vários eixos. Deslocamento aditivo (Translation).
`CTRANS(x, valor, y, valor, z, valor )`	Deslocamento aproximado para vários eixos. Deslocamento absoluto (Translation).
`x y z`	Deslocamento de ponto zero dos eixos (máx. 8)
`Valor`	Parte da translação

Fabricante da máquina

Com o MD18600 $MN_ MM_FRAME_FINE_TRANS se pode configurar o deslocamento fino nas seguintes variantes:

0:
O deslocamento fino não pode ser especificado nem programado. G58 e G59 não são possíveis.

1:
O deslocamento fino pode ser especificado e programado para Frames ajustáveis, Frames básicos, Frames programáveis, G58 e G59.

Descrição

Um deslocamento fino modificado pela operação HMI somente é ativado após a ativação do Frame correspondente, isto é, a ativação é realizada através do G500, G54...G599. Um deslocamento fino ativado de um Frame permanece ativado enquanto o Frame também estiver ativado.

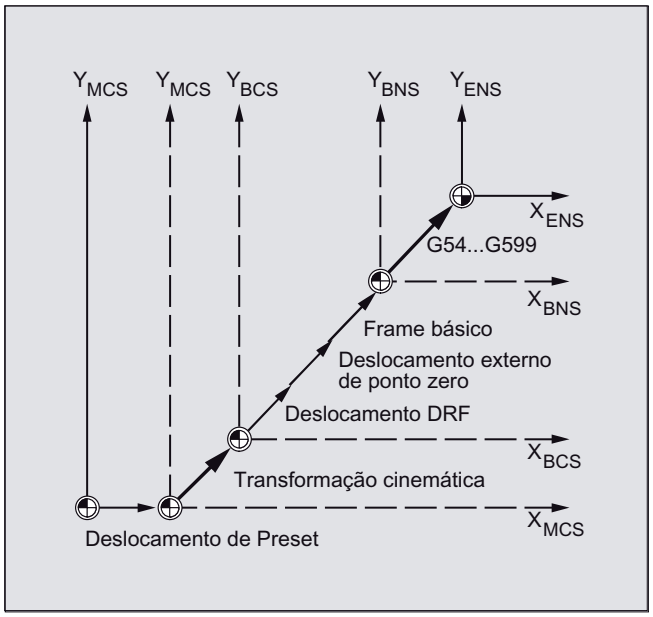
Nenhum componente de deslocamento contém o Frame programável. Se para o Frame programável for atribuído um Frame com deslocamento fino, então seu deslocamento total é formado a partir da soma dos deslocamentos fino e aproximado. Na leitura do Frame programável o deslocamento fino sempre é zero.

5.4 Deslocamento externo de ponto zero

Função

Dessa forma temos mais uma opção disponível, a de deslocar o ponto zero entre o sistema de coordenadas básico e o da peça de trabalho.

Com o deslocamento externo de ponto zero somente se pode programar deslocamentos lineares.



Programação

A programação dos valores de deslocamento $AA_ETRANS é realizada através da especificação das variáveis de sistema específicas de eixo.

Atribuir valor de deslocamento

```
$AA_ETRANS[eixo]=RI
```

RI é a variável de cálculo de tipo REAL que contém o novo valor.

O deslocamento externo normalmente não é especificado no programa de peça, mas definido pelo PLC.

Indicação

O valor escrito no programa de peça é ativado se na interface VDI (interface NCU-PLC) estiver definido o respectivo sinal.

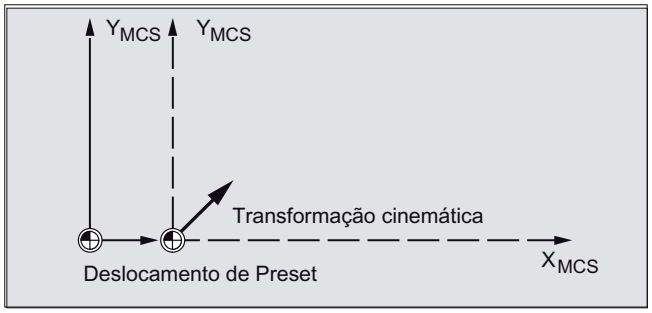
5.5 Deslocamento Preset (PRESETON)

Função

Para aplicações especiais pode ser necessário atribuir um novo valor real programado para um ou mais eixos na posição atual (estacionário).

⚠ CUIDADO

Com a função PRESETON o ponto de referência torna-se inválido. Por isso que esta função somente deve ser aplicada em eixos sem necessidade de ponto de referência. Se for necessário restabelecer o sistema original, então os pontos de referência deverão ser aproximados com G74, veja o capítulo "Gerenciamento de arquivos e programas".



Sintaxe

```
PRESETON(eixo, valor, ...)
```

Significado

`PRESETON`	Definir valor real
`Eixo`	Indicação do eixo da máquina
`Valor`	Novo valor real que deve ser aplicado no eixo especificado

Indicação

A definição do valor real com ações sincronizadas somente deveria ser realizado com a palavra-chave "WHEN" ou "EVERY".

Exemplo

A atribuição dos valores reais é realizada no sistema de coordenadas da máquina, os valores são relativos aos eixos da máquina.

```
N10 G0 A760
N20 PRESETON(A1,60)
```

O eixo A desloca-se até a posição 760. O eixo de máquina A1 contém o novo valor real 60 na posição 760. A partir de agora se posiciona no novo sistema de valor real.

5.6 Cálculo de Frame a partir de 3 pontos de medição no espaço (MEAFRAME)

Função

MEAFRAME é uma ampliação da linguagem 840D para dar suporte aos ciclos de medição.

A função MEAFRAME calcula o Frame a partir de três pontos ideais e dos pontos medidos correspondentes.

Se uma peça de trabalho é posicionada para a usinagem, sua posição relativa ao sistema de coordenadas cartesiano da máquina normalmente está deslocada e também girada em relação à sua posição ideal. Para a usinagem ou medição exata, torna-se necessário ou um ajuste físico dispendioso ou então a modificação dos movimentos no programa de peça.

Um Frame pode ser definido através do apalpamento de três pontos no espaço, cujas posições ideais são conhecidas. O apalpamento é realizado com um sensor de contato ou óptico, que entra especialmente em contato com os furos de precisão ou esferas de medição da placa de suporte.

Sintaxe

```
MEAFRAME IDEAL_POINT,MEAS_POINT,FIT_QUALITY)
```

Significado

`MEAFRAME`	Cálculo de Frame a partir de 3 pontos de medição no espaço	
`IDEAL_POINT`	campo real dimensional que contém as três coordenadas dos pontos ideais	
`MEAS_POINT`	campo real dimensional que contém as três coordenadas dos pontos medidos	
`FIT_QUALITY`	Variável Real,	com que se retornam as seguintes informações:
	-1:	Os pontos ideais estão próximos de uma reta: O Frame não pôde ser calculado. A variável Frame retornada contém um Frame neutro.
	-2:	Os pontos medidos estão próximos de uma reta: O Frame não pôde ser calculado. A variável Frame retornada contém um Frame neutro.
	-4:	O cálculo da matriz de rotação falhou por outro motivo
	valor positivo:	Soma das distorções (distâncias entre os pontos) que são necessárias para a transformar o triângulo medido em um triângulo ideal e congruente.

Indicação

Qualidade da medição

Para que as coordenadas medidas sejam atribuídas às coordenadas ideais com uma rotação/translação combinada, o triângulo formado pelos pontos de medição deverá ser congruente com o triângulo ideal. Isto se alcança com um algoritmo de compensação que minimiza a somatória do quadrado dos desvios que reformam o triângulo medido para o triângulo ideal.

A distorção efetiva necessária dos pontos de medição pode servir de indicador para a qualidade da medição e por isso é especificada como variável adicional do `MEAFRAME`.

Indicação

O Frame gerado pelo `MEAFRAME` pode ser transformado em outro Frame do encadeamento de Frames através da função `ADDFRAME`.
Veja o exemplo: Encadeamento de Frames "Encadeamento com ADDFRAME".

Para mais informações sobre os parâmetros do `ADDFRAME(FRAME, STRING)`, veja /FB1/ Manual de funções básicas; Eixos, Sistemas de coordenadas, Frames (K2), Capítulo "Encadeamento de FRAMEs".

Exemplo

Código de programa	Comentário
	; Programa de peça 1
DEF FRAME CORR_FRAME	

Definição de pontos de medição

Programação	Comentário
DEF REAL IDEAL_POINT[3,3] = SET(10.0,0.0,0.0, 0.0,10.0,0.0, 0.0,0.0,10.0)	
DEF REAL MEAS_POINT[3,3] = SET (10.1,0.2,-0.2, -0.2,10.2,0.1, -0.2,0.2,9.8)	; para teste
DEF REAL FIT_QUALITY = 0	
DEF REAL ROT_FRAME_LIMIT = 5	; permite no máx. 5 graus de rotação da posição da peça
DEF REAL FIT_QUALITY_LIMIT = 3	; permite no máx. 3 mm de deslocamento entre o triângulo ideal e o triângulo medido
DEF REAL SHOW_MCS_POS1[3]	
DEF REAL SHOW_MCS_POS2[3]	
DEF REAL SHOW_MCS_POS3[3]	

```
Código de programa                                  Comentário
N100 G01 G90 F5000
N110 X0 Y0 Z0
N200 CORR_FRAME=MEAFRAME(IDEAL_POINT,MEAS
_POINT,FIT_QUALITY)

N230 IF FIT_QUALITY < 0
SETAL(65000)
GOTOF NO_FRAME
ENDIF
N240 IF FIT_QUALITY > FIT_QUALITY_LIMIT
SETAL(65010)
GOTOF NO_FRAME
ENDIF
N250 IF CORR_FRAME[X,RT] > ROT_FRAME_LIMIT      ;  Limitação do 1° ângulo RPY
SETAL(65020)
GOTOF NO_FRAME
ENDIF
N260 IF CORR_FRAME[Y,RT] > ROT_FRAME_LIMIT      ;  Limitação do 2° ângulo RPY
SETAL(65021)
GOTOF NO_FRAME
ENDIF
N270 IF CORR_FRAME[Z,RT] > ROT_FRAME_LIMIT      ;  Limitação do 3° ângulo RPY
SETAL(65022)
GOTOF NO_FRAME
ENDIF
N300 $P_IFRAME=CORR_FRAME                       ;  Ativar Frame de apalpamento com
                                                   um Frame de definição
                                                ;  Verificar Frame com o
                                                   posicionamento dos eixos
                                                   geométricos nos pontos ideais
N400 X=IDEAL_POINT[0,0] Y=IDEAL_POINT[0,1]
Z=IDEAL_POINT[0,2]
N410 SHOW_MCS_POS1[0]=$AA_IM[X]
N420 SHOW_MCS_POS1[1]=$AA_IM[Y]
N430 SHOW_MCS_POS1[2]=$AA_IM[Z]
N500 X=IDEAL_POINT[1,0] Y=IDEAL_POINT[1,1]
Z=IDEAL_POINT[1,2]
N510 SHOW_MCS_POS2[0]=$AA_IM[X]
N520 SHOW_MCS_POS2[1]=$AA_IM[Y]
N530 SHOW_MCS_POS2[2]=$AA_IM[Z]
N600 X=IDEAL_POINT[2,0] Y=IDEAL_POINT[2,1]
Z=IDEAL_POINT[2,2]
N610 SHOW_MCS_POS3[0]=$AA_IM[X]
```

```
Código de programa                                  Comentário
N620 SHOW_MCS_POS3[1]=$AA_IM[Y]
N630 SHOW_MCS_POS3[2]=$AA_IM[Z]
N700 G500                                           ; Desativar Frame de definição,
                                                      definido com Frame zero (sem
                                                      valor especificado).
No_FRAME                                            ; Desativar Frame de definição,
                                                      definido com Frame zero (sem
                                                      valor especificado)
M0
M30
```

Exemplo de encadeamento de Frames

Encadeamento de MEAFRAME para correções

A função `MEAFRAME( )` fornece um Frame de correção. Se este Frame de correção for encadeado com o Frame ajustável `$P_UIFR[1]` que estava ativo na chamada da função, p. ex. `G54`, então obtemos um Frame ajustável para outras conversões para movimentos ou usinagem.

Encadeamento com ADDFRAME

Se este Frame de correção deve atuar em outra posição do encadeamento de Frames, ou se antes do Frame ajustável ainda existem outros Frames ativos, então se pode utilizar a função `ADDFRAME( )` para concatenar um dos Frames básicos de canal ou um Frame de sistema.

Neste caso, no Frame não podem estar ativos:

- Espelhamento com `MIRROR`
- Escala com `SCALE`

Os parâmetros de entrada para valores nominais e reais são as coordenadas da peça de trabalho. No sistema básico do comando estas coordenadas sempre são especificadas como medidas

- métricas ou em polegadas `(G71/G70)` e como
- relativas ao raio `(DIAMOF)`.

5.7 Frames globais da NCU

Função

Para cada NCU os Frames globais da NCU ocorre apenas uma vez em todos canais. Os Frames globais da NCU podem ser gravados e lidos a partir de todos canais. A ativação dos Frames globais da NCU é realizada no respectivo canal.

Através dos Frames globais os **eixos de canal e eixos da máquina** podem ter escala alterada e serem espelhados com os deslocamentos.

Relações geométricas e encadeamento de Frames

Para Frames globais não existe nenhuma relação geométrica entre os eixos. Por isso que não pode ser executada nenhuma rotação e nenhuma programação de identificadores de eixos geométricos.

- As rotações não podem ser aplicadas em Frames globais. A programação de uma rotação é negada com o alarme: "18310 canal %1 bloco %2 Frame: Rotação não permitida".
- A realização do encadeamento de Frames globais e Frames específicos de canal é possível. O Frame resultante contém todos componentes de Frame, inclusive as rotações de todos eixos. A atribuição de um Frame com componentes de rotação em um Frame global é negada com o alarme "Frame: Rotação não permitida".

Frames básicos globais da NCU

Frames básicos globais da NCU $P_NCBFR[n]

Podem ser configurados até 8 Frames básicos globais da NCU:

Ao mesmo tempo podem existir Frames básicos específicos de canal.

Frames globais podem ser gravados e lidos a partir de todos canais de uma NCU. O usuário deve providenciar uma coordenação de canais ao gravar Frames globais. Por exemplo, isso pode ser realizado por marcadores Wait `(WAITMC)`.

Fabricante da máquina

O número de Frames básicos globais é configurado através de dados de máquina, veja /FB1/ Manual de funções básicas; Eixos, Sistemas de coordenadas, Frames (K2).

Frames ajustáveis globais da NCU $P_UIFR[n]

Todos os Frames ajustáveis `G500`, `G54...G599` podem ser configurados como globais da NCU ou específicos de canal.

Fabricante da máquina

Todos os Frames ajustáveis podem ser reconfigurados em Frames globais com o auxílio de um dado de máquina $MN_MM_NUM_GLOBAL_USER_FRAMES.

Como identificador de eixo nos comandos de programação de Frames podem ser utilizados identificadores específicos de canal e identificadores de eixo de máquina. A programação dos identificadores de eixos geométricos é negada com um alarme.

5.7.1 Frames específicos de canal ($P_CHBFR, $P_UBFR)

Função

Frames ajustáveis ou Frames básicos podem,

- através do programa de peça e
- através do BTSS

serem gravados e lidos pela interface de operação como HMI Advanced e pelo PLC.

O deslocamento fino também é possível nos Frames globais. A supressão do Frames globais é realizada da mesma forma como dos Frames específicos de canal, através do `G53`, `G153`, `SUPA` e `G500`.

Fabricante da máquina

Através do MD28081 MM_NUM_BASE_FRAMES se pode configurar o número de Frames básicos no canal. A configuração padrão está definida de modo que exista pelo menos um Frame básico por canal. No máximo são possíveis 8 Frames básicos. Além dos 8 Frames básicos no canal ainda podem existir 8 Frames básicos globais da NCU.

Frames específicos de canal

$P_CHBFR[n]

Através da variável de sistema `$P_CHBFR[n]` se pode ler e gravar os Frames básicos. Ao se gravar um Frame básico, o Frame básico total encadeado não é ativado, a ativação somente ocorre com a execução de uma instrução `G500`, `G54...G599`. A variável serve especialmente como memória de gravações no Frame básico da HMI ou PLC. Estas variáveis Frame são salvas através do backup de dados.

Primeiro Frame básico no canal

Uma gravação na variável pré-definida `$P_UBFR` não ativa simultaneamente o Frame básico com o índice de campo 0, a ativação somente é realizada com a execução de uma instrução `G500`, `G54...G599`. A variável também pode ser gravada e lida no programa.

$P_UBFR

`$P_UBFR` é idêntico ao `$P_CHBFR[0]`. Como padrão sempre existe um Frame básico no canal, de modo que a variável de sistema seja compatível com versões anteriores. Se não existir um Frame básico específico de canal, durante a gravação ou leitura será emitido o alarme "Frame: Instrução inválida".

5.7.2 Frames ativos no canal

Função

Os Frames ativos no canal são especificados pelo programa de peça através das respectivas variáveis de sistema destes Frames. Aqui também pertencem os Frames de sistema. Através destas variáveis de sistema pode-se ler e gravar o atual Frame de sistema no programa de peça.

Atuais Frames ativos no canal

Vista geral

Atuais Frames de sistema	para:
$P_PARTFRAME	TCARR e PAROT
$P_SETFRAME	Definir valor real e contato de referência
$P_EXTFRAME	Deslocamento externo de ponto zero
$P_NCBFRAME[n]	Atuais Frames básicos globais da NCU
$P_CHBFRAME[n]	Atuais Frames básicos de canal
$P_BFRAME	Atual 1º Frame básico no canal
$P_ACTBFRAME	Frame básico total
$P_CHBFRMASK e $P_NCBFRMASK	Frame básico total
$P_IFRAME	Atual Frame ajustável
Atuais Frames de sistema	para:
$P_TOOLFRAME	TOROT e TOFRAME
$P_WPFRAME	Pontos de referência de peça
$P_TRAFRAME	Transformações
$P_PFRAME	Atual Frame programável
Atual Frame de sistema	para:
$P_CYCFRAME	Ciclos
P_ACTFRAME	Atual Frame total
Encadeamento de FRAMEs	O atual Frame é formado pelo Frame básico total

$P_NCBFRAME[n] Atuais Frames básicos globais da NCU

Através da variável de sistema `$P_NCBFRAME[n]` podem ser lidos e gravados os atuais elementos de campo de Frames básicos globais. O Frame básico total resultante é processado pelo processo de gravação no canal.

O Frame alterado somente estará ativo no canal em que o Frame foi programado. Se o Frame deve ser alterado em todos canais de uma NCU, então $P_NCBFR[n] e $P_NCBFRAME[n] devem ser gravados simultaneamente. Os outros canais ainda devem ativar o Frame, por exemplo com G54. Ao gravar um Frame básico, será recalculado o Frame básico total.

$P_CHBFRAME[n] Atuais Frames básicos de canal

Através da variável de sistema `$P_CHBFRAME[n]` podem ser lidos e gravados os atuais elementos de campo de Frame básico de canal. O Frame básico total resultante é processado pelo processo de gravação no canal. Ao gravar um Frame básico, será recalculado o Frame básico total.

$P_BFRAME Atual 1º Frame básico no canal

Através da variável Frame pré-definida `$P_BFRAME` o atual Frame básico com o índice de campo 0 válido no canal, poderá ser lido e gravado no programa de peça. O Frame básico gravado é imediatamente processado.

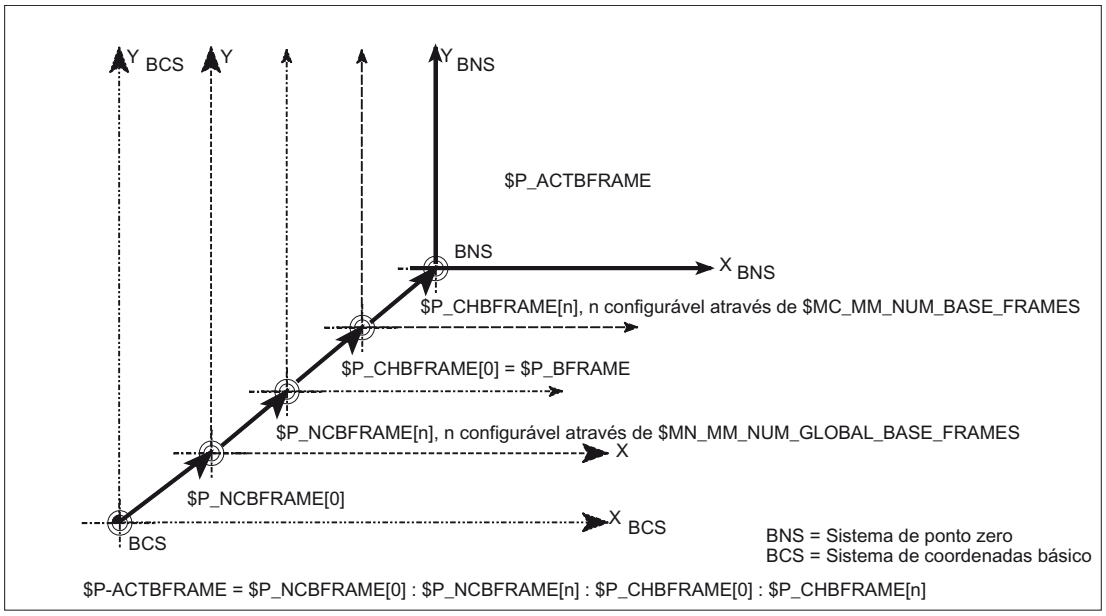
`$P_BFRAME` é idêntico ao `$P_CHBFRAME[0]`. Como padrão a variável de sistema sempre tem um valor válido. Se não existir um Frame básico específico de canal, durante a gravação ou leitura será emitido o alarme "Frame: Instrução inválida".

$P_ACTBFRAME Frame básico total

A variável `$P_ACTBFRAME` determina o Frame básico total encadeado. A variável somente pode ser lida.

`$P_ACTFRAME` corresponde ao

`$P_NCBFRAME[0]` : ... : `$P_NCBFRAME[n]` : `$P_CHBFRAME[0]` : ... : `$P_CHBFRAME[n]`.



$P_CHBFRMASK e $P_NCBFRMASK Frame básico total

Através das variáveis de sistema `$P_CHBFRMASK` e `$P_NCBFRMASK` o usuário pode escolher quais Frames básicos ele quer adicionar ao cálculo do Frame básico "total". As variáveis somente podem ser programadas uma vez no programa e lidas através do BTSS. O valor das variáveis é interpretado como máscara de Bit e indica qual elemento de campo do Frame básico de `$P_ACTFRAME` entra no cálculo.

Com o `$P_CHBFRMASK` se pode especificar quais Frames básicos específicos de canal, e com `$P_NCBFRMASK`quais Frames básicos globais da NCU serão processados.

Com a programação das variáveis é recalculado o Frame básico total e o Frame total. Após Reset e no ajuste básico o valor de

`$P_CHBFRMASK = $MC_CHBFRAME_RESET_MASK` e

`$P_NCBFRMASK = $MC_CHBFRAME_RESET_MASK`.

p. ex.

`$P_NCBFRMASK = 'H81'` ;$P_NCBFRAME[0] : $P_NCBFRAME[7]

`$P_CHBFRMASK = 'H11'` ;$P_CHBFRAME[0] : $P_CHBFRAME[4]

$P_IFRAME Atual Frame ajustável

Através da variável Frame pré-definida `$P_IFRAME`, o atual Frame ajustável, o que estiver válido no canal, pode ser lido e gravado no programa de peça. O Frame ajustável que está gravado é imediatamente processado.

No caso dos Frames ajustáveis globais da NCU, o Frame alterado apenas atua no canal em que foi programado. Se o Frame de todos canais de uma NCU deve ser alterado, então o `$P_UIFR[n]` e o `$P_IFRAME` devem ser gravados simultaneamente. Os outros canais ainda devem ativar o Frame correspondente, por exemplo com G54.

$P_PFRAME Atual Frame programável

O `$P_PFRAME` é o Frame programável resultante da programação do `TRANS/ATRANS`, `G58/G59`, `ROT/AROT`, `SCALE/ASCALE`, `MIRROR/AMIRROR` e da atribuição do `CTRANS`, `CROT`, `CMIRROR`, `CSCALE` no FRAME programável.

Atual variável Frame programável que forma a relação entre o ajustável

- Sistema de ponto zero (ENS) e o
- Sistema de coordenadas da peça (WCS).

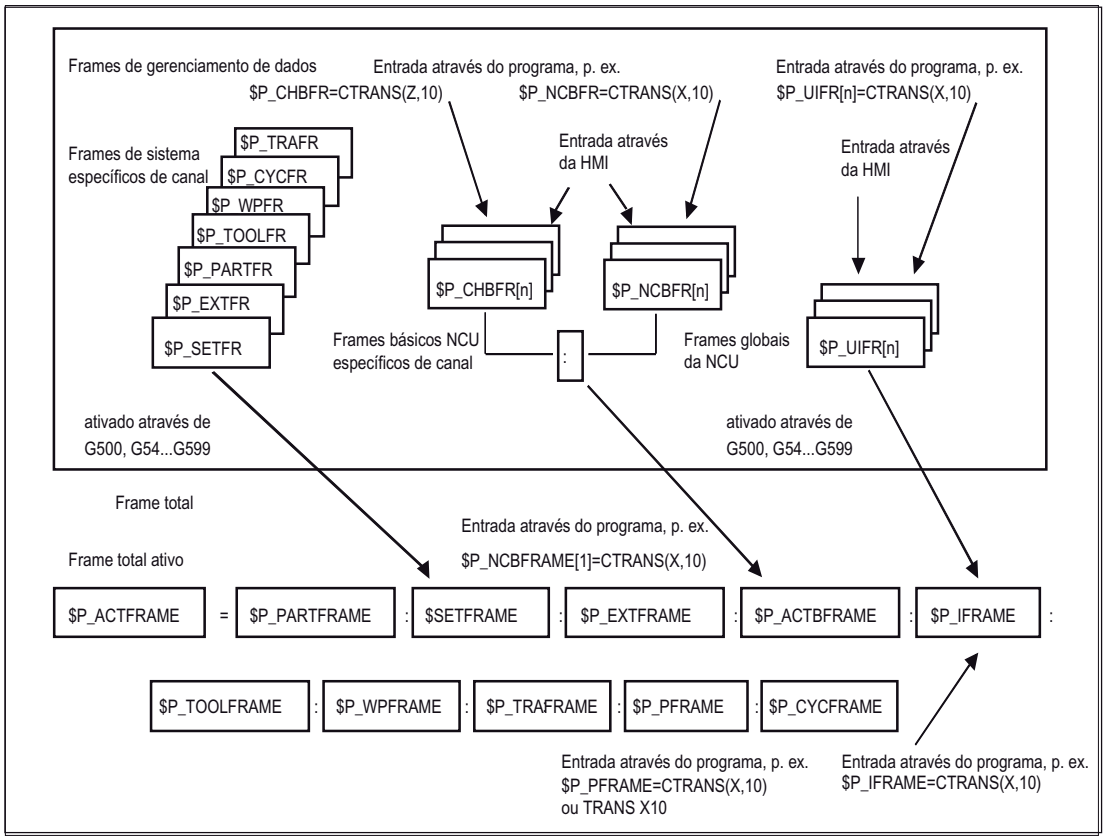
P_ACTFRAME Atual Frame total

O atual Frame total $P_ACTFRAME resultante agora resulta como encadeamento de todos Frames básicos, do atual Frame ajustável e do Frame programável. O atual Frame sempre será atualizado quando um componente do Frame for alterado.

`$P_ACTFRAME` corresponde ao

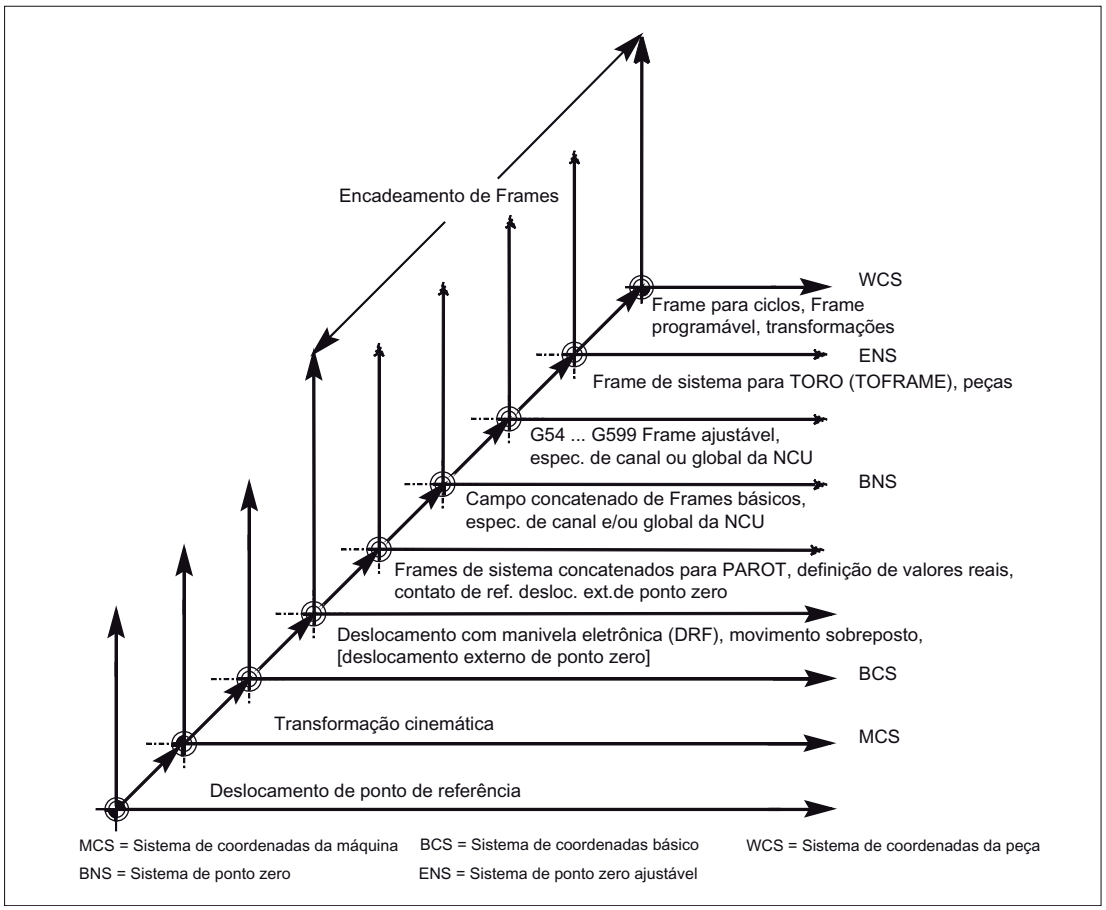
`$P_PARTFRAME : $P_SETFRAME : $P_EXTFRAME : $P_ACTBFRAME : $P_IFRAME :`

`$P_TOOLFRAME : $P_WPFRAME : $P_TRAFRAME : $P_PFRAME : $P_CYCFRAME`



Encadeamento de Frames

O atual Frame é formado a partir do Frame básico total, do Frame ajustável, do Frame de sistema e do Frame programável conforme o atual Frame total acima mencionado.


Encadeamento de Frames
WCS
Frame para ciclos, Frame programável, transformações
ENS
Frame de sistema para TORO (TOFRAME), peças
G54 ... G599 Frame ajustável, espec. de canal ou global da NCU
BNS
Campo concatenado de Frames básicos, espec. de canal e/ou global da NCU
Frames de sistema concatenados para PAROT, definição de valores reais, contato de ref. desloc. ext.de ponto zero
Deslocamento com manivela eletrônica (DRF), movimento sobreposto, [deslocamento externo de ponto zero]
BCS
Transformação cinemática
MCS
Deslocamento de ponto de referência
MCS = Sistema de coordenadas da máquina
BCS = Sistema de coordenadas básico
WCS = Sistema de coordenadas da peça
BNS = Sistema de ponto zero
ENS = Sistema de ponto zero ajustável

Transformações 6

6.1 Programação geral de tipos de transformação

Função geral

Para adaptação do comando em diversas cinemáticas de máquina existe a possibilidade de se programar a seleção dos tipos de transformações com parâmetros adequados. Através destes parâmetros pode ser acordado para a transformação selecionada tanto a orientação da ferramenta no espaço como os movimentos da orientação dos eixos rotativos.

Em transformações de três, quatro e cinco eixos, as indicações de posição programadas sempre estão relacionadas à ponta da ferramenta que é acompanhada de forma ortogonal à superfície de usinagem disponível no espaço. As coordenadas cartesianas são convertidas do sistema de coordenadas básico para o sistema de coordenadas da máquina e estão relacionadas aos eixos geométricos. Elas descrevem o ponto de trabalho. Os eixos rotativos virtuais descrevem as orientações da ferramenta no espaço e são programadas com TRAORI.

Na transformação cinemática as posições podem ser programadas no sistema de coordenadas cartesiano. O comando numérico transforma os movimentos de deslocamento do sistema de coordenadas cartesiano programados com TRANSMIT, TRACYL e TRAANG em movimentos de deslocamento dos eixos reais da máquina.

Programação

Transformações de três, quatro e cinco eixos TRAORI

A transformação de orientação acordada é ativada com o comando TRAORI e com os três possíveis parâmetros para número de transformação, vetor de orientação e Offset de eixo rotativo.

```
TRAORI(número de transformação, vetor de orientação, Offsets de eixo rotativo)
```

Transformações cinemáticas

Às transformações cinemáticas pertencem as transformações acordadas `TRANSMIT(número de transformação)`

```
TRACYL(diâmetro de trabalho, número de transformação)
TRAANG(ângulo do eixo inclinado, número de transformação)
```

Desativar a transformação ativa

Com `TRAFOOF` se desativa a transformação que estiver ativa no momento.

Transformação de orientação

Transformações de três, quatro e cinco eixos TRAORI

Para uma usinagem otimizada de superfícies de formas espaciais na área de trabalho da máquina, as máquinas-ferramenta requerem eixos adicionais aos três eixos lineares X, Y e Z. Os eixos adicionais descrevem a orientação no espaço e a seguir serão chamados de eixos de orientação. Eles estão disponíveis como eixos de rotação em quatro tipos de máquina de diferentes cinemáticas.

1. Cabeçote orientável de dois eixos, p. ex. cabeçote de ferramentas cardânico com um eixo rotativo paralelo à um eixo linear e com mesa fixa de ferramenta.
2. Mesa giratória de dois eixos, p. ex. cabeçote orientável fixo com mesa giratória de ferramenta e dois eixos.
3. Cabeçote orientável de um eixo e mesa giratória de um eixo, p. ex. um cabeçote giratório com ferramenta rotativa com mesa giratória de ferramenta em torno de um eixo.
4. Cabeçote orientável de dois eixos e mesa giratória de um eixo, p. ex. uma mesa giratória de ferramenta em torno de um eixo e um cabeçote giratório com ferramenta rotativa em torno de si mesmo.

As **transformações de 3 e 4 eixos** são formas especiais da transformação de 5 eixos e são programadas de forma análoga às transformações de 5 eixos.

A "**transformação genérica de 3, 4, 5, 6 eixos**", com suas funções, cobre os eixos rotativos dispostos perpendicularmente e as transformações para o cabeçote de fresar cardânico, e pode ser ativada com TRAORI para qualquer outra transformação de orientação, assim como para estes quatro tipos de máquina. Na transformação genérica de 5 e 6 eixos, a orientação da ferramenta possui mais um terceiro grau de liberdade, onde a ferramenta pode ser girada em torno de seu próprio eixo em qualquer direção dentro do espaço.

Literatura: /FB3/ Manual de funções especiais; Transformações de 3 a 5 eixos (F2)

Posição básica da orientação da ferramenta independente da cinemática

ORIRESET

Se uma transformação de orientação estiver ativa com TRAORI, então com ORIRESET se pode especificar as posições básicas de até 3 eixos de orientação com os parâmetros opcionais A, B e C. A associação da seqüência dos parâmetros programados com os eixos rotativos é realizada de acordo com a seqüência dos eixos de orientação definida através da transformação. A programação do ORIRESET(A, B, C) faz com que os eixos de orientação se desloquem de modo linear e sincronizado de sua posição momentânea até a posição básica especificada.

Transformações cinemáticas

TRANSMIT e TRACYL

No caso das operações de fresamento em tornos, para a transformação acordada se pode

1. programar uma usinagem frontal na fixação rotativa com TRANSMIT ou
2. uma usinagem de um número qualquer de ranhuras em corpos cilíndricos com TRACYL.

TRAANG

Por exemplo, se o eixo de penetração da tecnologia de retificação também deve penetrar de modo inclinado, então se pode programar um ângulo parametrizável com TRAANG para a transformação acordada.

Deslocamento PTP cartesiano

O "Deslocamento PTP cartesiano" também pertence às transformações cinemáticas, onde se pode programar até 8 diferentes posições articuladas STAT= . As posições são programadas no sistema de coordenadas cartesiano, onde o movimento da máquina é realizado em coordenadas da máquina.

Literatura:
/FB2/ Manual de funções ampliadas; Transformação cinemática (M1)

Transformações encadeadas

Sempre podem ser ativadas duas transformações sucessivamente. Na segunda transformação encadeada desse modo se assumem as partes de movimentos dos eixos da primeira transformação.

Como primeira transformação são possíveis:

- Transformação de orientação TRAORI
- Transformação polar TRANSMIT
- Transformação cilíndrica TRACYL
- Transformação de eixo inclinado TRAANG

A segunda transformação deve ser de eixo inclinado TRAANG

6.1.1 Movimentos de orientação nas transformações

Movimentos de deslocamento e movimentos de orientação

Os movimentos de deslocamentos das orientações programáveis dependem primeiramente do tipo de máquina. Na transformação de três, quatro e cinco eixos com TRAORI são os eixos rotativos ou os eixos lineares orientáveis que descrevem os movimentos de orientação da ferramenta.

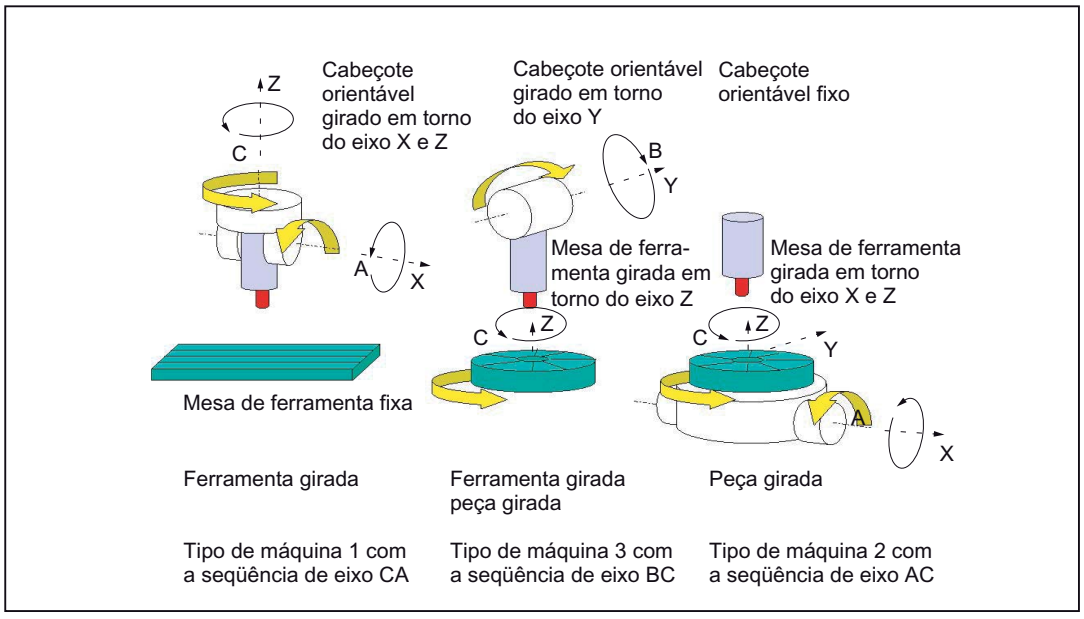
As alterações nas posições dos eixos rotativos que participam na transformação da orientação geram movimentos de compensação nos demais eixos da máquina. A posição da ponta da ferramenta permanece inalterada com isso.

Os movimentos de orientação da ferramenta podem ser programados através dos identificadores de eixo rotativo A…, B…, C… dos eixos virtuais, e dependendo da aplicação, através da indicação de ângulos eulerianos e ângulos RPY ou através de vetores de direção e vetores normais de superfície, vetores normalizados para o eixo de rotação de um cone ou para orientação intermediária em uma superfície periférica cônica.

Na transformação cinemática com TRANSMIT, TRACYL e TRAANG o comando numérico transforma os movimentos de deslocamento programados do sistema de coordenadas cartesiano em movimentos de deslocamento dos eixos reais da máquina.

Cinemática da máquina na transformação de três, quatro e cinco eixos TRAORI

Se pode girar a ferramenta ou a mesa de ferramenta mediante até dois eixos rotativos. Uma combinação de um cabeçote orientável de um eixo e a mesa giratória também é possível.


Cabeçote orientável girado em torno do eixo X e Z
Cabeçote orientável girado em torno do eixo Y
Cabeçote orientável fixo
Z
C
A
X
B
Y
Mesa de ferramenta girada em torno do eixo Z
Mesa de ferramenta girada em torno do eixo X e Z
Mesa de ferramenta fixa
Ferramenta girada
Ferramenta girada peça girada
Peça girada
Tipo de máquina 1 com a seqüência de eixo CA
Tipo de máquina 3 com a seqüência de eixo BC
Tipo de máquina 2 com a seqüência de eixo AC

Tipo de máquina	Programação da orientação
Transformação de três eixos para tipos de máquina 1 e 2	Programação da orientação de ferramenta apenas no plano que estiver perpendicular ao eixo rotativo. Existem **dois** eixos translatórios (eixos lineares) e **um** eixo rotatório (eixo rotativo).
Transformação de quatro eixos para tipos de máquina 1 e 2	Programação da orientação de ferramenta apenas no plano que estiver perpendicular ao eixo rotativo. Existem **três** eixos translatórios (eixos lineares) e **um** eixo rotatório (eixo rotativo).
Transformação de cinco eixos para tipo de máquina 3 Cabeçote orientável de um eixo e mesa giratória de um eixo	Programação da transformação de orientação. Cinemática com **três** eixos lineares e **dois** eixos rotativos ortogonais. Os eixos rotativos são paralelos à dois dos três eixos lineares. O primeiro eixo rotativo é movimentado por dois eixos lineares cartesianos. Ele gira o terceiro eixo linear com a ferramenta. O segundo eixo rotativo gira a peça de trabalho.

Transformações genéricas de 5 e 6 eixos

Tipo de máquina	Programação da transformação de orientação
Transformação genérica de cinco e seis eixos para tipo de máquina 4 Cabeçote orientável de dois eixos com ferramenta rotativa em torno de si mesma e mesa giratória de um eixo	Programação da transformação de orientação. Cinemática com **três** eixos lineares e **três** eixos rotativos ortogonais. Os eixos rotativos são paralelos à dois dos três eixos lineares. O primeiro eixo rotativo é movimentado por dois eixos lineares cartesianos. Ele gira o terceiro eixo linear com a ferramenta. O segundo eixo rotativo gira a peça de trabalho. A orientação básica da ferramenta pode ser programada através de um giro adicional em torno de si mesma com o ângulo de giro THETA.

Na chamada da "Transformação genérica de três, quatro e cinco eixos" também se pode passar a orientação básica da ferramenta. Não se pode aplicar mais as restrições relacionadas às direções dos eixos rotativos. Se os eixos rotativos não estiverem exatamente perpendiculares entre si ou se os eixos rotativos existentes não estiverem exatamente paralelos com os eixos lineares, a "Transformação genérica de cinco e seis eixos" pode oferecer melhores resultados de orientação da ferramenta.

Transformações cinemáticas TRANSMIT, TRACYL e TRAANG

Para operações de fresamento em tornos ou para um eixo que avança inclinado na operação de retificação se aplicam as seguintes disposições de eixo (em função da transformação em casos padrão):

TRANSMIT	Ativação da transformação polar
usinagem frontal na fixação rotativa	um eixo rotativo um eixo de penetração perpendicular ao eixo de rotação um eixo longitudinal paralelo ao eixo de rotação

TRACYL	Ativação da transformação de superfície cilíndrica
Usinagem de qualquer ranhura que percorre no corpo cilíndrico	um eixo rotativo um eixo de penetração perpendicular ao eixo de rotação um eixo longitudinal paralelo ao eixo de rotação

TRAANG	Ativação da transformação de eixos inclinados
Usinagem com eixo de penetração inclinado	um eixo rotativo um eixo de penetração com ângulo parametrizável um eixo longitudinal paralelo ao eixo de rotação

Deslocamento PTP cartesiano

O movimento da máquina é realizado em coordenadas da máquina e se programa com:

TRAORI	Ativação da transformação
Deslocamento ponto a ponto PTP	Aproximação da posição em sistema de coordenadas cartesiano (MCS)
CP	Movimento de percurso dos eixos cartesianos em (BCS)
STAT	A posição das articulações está em função da transformação
TU	O ângulo com que os eixos percorrem o curso mais curto

Deslocamento PTP na transformação de 5 e 6 eixos

O movimento da máquina é realizado em coordenadas da máquina e a orientação da ferramenta podem tanto ser programada com posições de eixo rotativo como com vetores eulerianos e vetores RPY independentes de cinemática ou com vetores de direção.

Desse modo é possível realizar a interpolação de eixos rotativos, interpolação de vetores com interpolação de grande circunferência ou interpolação do vetor de orientação sobre uma superfície periférica cônica.

Exemplo de transformação de três a cinco eixos em um cabeçote de fresar cardânico

A máquina-ferramenta possui pelo 5 eixos, destes

- Três eixos translatórios para movimentos em linhas retas, que movimentam o ponto de trabalho em qualquer posição na área de trabalho.
- Dois eixos orientáveis rotativos que estão posicionados conforme um ângulo projetável (frequentemente 45 graus) permitem a realização da orientação da ferramenta no espaço, restrita na posição de 45 graus sobre uma meia esfera.

6.1.2 Vista geral da transformação de orientação TRAORI

Possíveis tipos de programação em conjunto com o TRAORI

Tipo de máquina	Programação com transformação TRAORI ativa
Tipos de máquina 1, 2 ou 3 Cabeçote orientável de dois eixos ou mesa giratória de dois eixos ou uma combinação de cabeçote orientável de um eixo e mesa giratória de um eixo.	A seqüência de eixos de orientação e o sentido de orientação da ferramenta podem ser projetados **relacionados à máquina** através de dados de máquina e em função da cinemática da máquina ou projetados **relacionados à ferramenta** com orientação programável e independente da cinemática da máquina Os sentidos de giro dos eixos de orientação no sistema de referência são programados com: - Sistema de referência ORIMKS = Sistema de coordenadas da máquina - Sistema de referência ORIWKS = Sistema de coordenadas da peça O ajuste básico é ORIWKS. Programação dos eixos de orientação com: A, B, C direto das posições de eixo da máquina A2, B2, C2 programação de ângulos de eixos virtuais com - ORIEULER através de ângulo euleriano (Standard) - ORIRPY através de ângulo RPY - ORIVIRT1 através de eixos de orientação virtuais de 1ª definição - ORIVIRT2 através de eixos de orientação virtuais de 2ª definição com diferença no tipo de interpolação: **Interpolação linear** - ORIAXES de eixos de orientação dos eixos da máquina **Interpolação de grande circunferência** (interpolação do vetor de orientação) - ORIVECT de eixos de orientação Programação dos eixos de orientação através da indicação A3, B3, C3 dos componentes de vetor (normal de sentido/superfície) Programação da orientação de ferramenta resultante A4, B4, C4 do vetor normal de superfície no início do bloco A5, B5, C5 do vetor de normal de superfície no fim do bloco Ângulo de avanço LEAD para orientação da ferramenta Ângulo lateral TILT para orientação de ferramenta

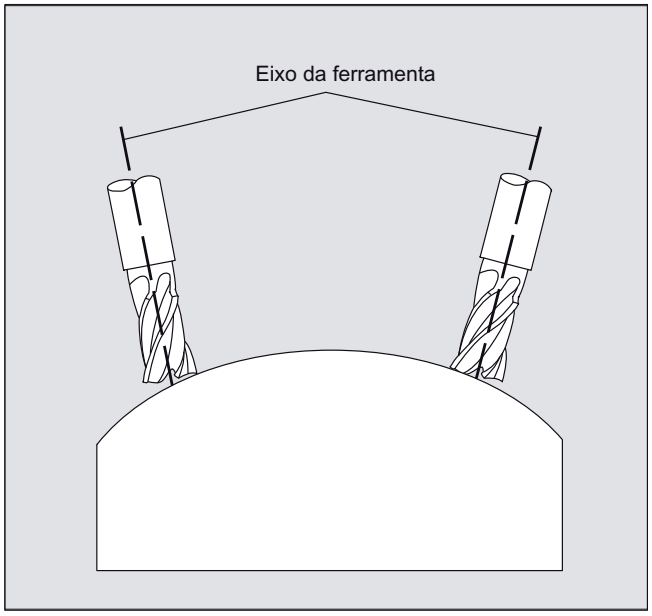
Tipo de máquina	Programação com transformação TRAORI ativa
	Interpolação do vetor de orientação sobre uma superfície periférica cônica Alterações de orientação sobre uma superfície periférica cônica **qualquer no espaço** através da interpolação: - ORIPLANE no plano (interpolação de grande circunferência) - ORICONCW sobre uma superfície periférica cônica em sentido horário - ORICONCCW sobre uma superfície periférica cônica em sentido anti-horário Vetores de direção A6, B6, C6 (eixo de rotação do cone) - Interpolação OICONIO sobre uma superfície periférica cônica com: Vetores intermediários A7, B7, C7 (orientação inicial e final) ou - ORICONTO sobre uma superfície periférica cônica de transição tangencial Alterações de orientação relacionadas **à uma trajetória** com - Especificação ORICURVE do movimento de dois pontos de contato através de Polinômios de orientação de até 5º grau PO[XH]=(xe, x2, x3, x4, x5) Polinômios de orientação de até 5º grau PO[YH]=(ye, y2, y3, y4, y5) Polinômio de orientação de até 5º grau PO[ZH]=(ze, z2, z3, z4, z5) - A suavização do decurso de orientação ORIPATHS com fase de reorientação A8, B8, C8 da ferramenta corresponde ao(à): Direção e distância de curso da ferramenta no movimento de retração
Tipos de máquina 1 e 3 Os demais tipos de máquina com giro adicional da ferramenta em torno de si mesma requerem um 3º eixo rotativo Transformação de orientação como na transformação genérica de 6 eixos. Rotações do vetor de orientação.	Programação **das rotações** da orientação da ferramenta com ângulo de avanço LEAD relativo ao vetor normal de superfície PO[PHI] Programação de um polinômio de até 5º grau TILT Ângulo lateral da rotação em torno da tangente da trajetória (sentido Z) PO[PSI] Programação de um polinômio de 5º grau THETA Ângulo de rotação (rotação em torno da direção da ferramenta em Z) THETA= Valor a ser alcançado no final do bloco THETA=AC(...) Comutar por blocos para dimensões absolutas THETA=IC(...) Comutar por blocos para dimensões incrementais THETA=Θ_e Interpolar ângulo programado G90/G91 PO[THT]=(..) Programação de um polinômio de 5º grau Programação do vetor de rotação - ORIROTA Rotação absoluta - ORIROTR Vetor de rotação relativo - ORIROTT Vetor de rotação tangencial
Orientação relativa à trajetória para alterações de orientação relativas à trajetória ou rotação do vetor de rotação tangencial à trajetória	Alterações de orientação **relativas à trajetória** com - Orientação de ferramenta ORIPATH relativa à trajetória - ORIPATHS adicionalmente em uma dobra no decurso da orientação Programação do vetor de rotação - Vetor de rotação tangencial ORIROTC, rotação com a tangente da trajetória

6.2 Transformação de três, quatro e cinco eixos (TRAORI)

6.2.1 Relações gerais do cabeçote de ferramenta cardânico

Função

Para obter as condições ideais de corte na usinagem de superfícies curvadas, se deve permitir a variação do ângulo de ataque da ferramenta.

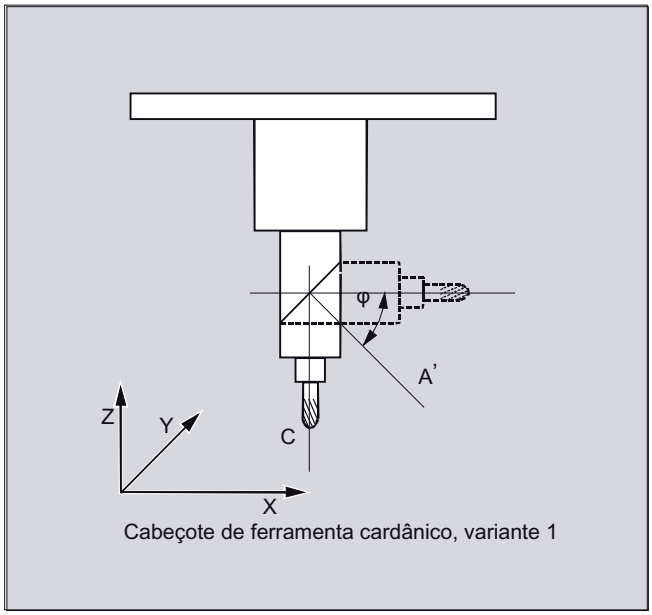


Com qual forma construtiva de máquina isso é obtido, está armazenado nos dados do eixo.

Transformação de 5 eixos

Cabeçote de ferramenta cardânico

Aqui estão definidos três eixos lineares (X, Y, Z) e dois eixos de orientação (C, A) do ângulo de ataque e ponto de trabalho da ferramenta. Um dos dois eixos de orientação está definido como eixo inclinado, aqui no exemplo A', como na maioria dos casos, na disposição de 45°.

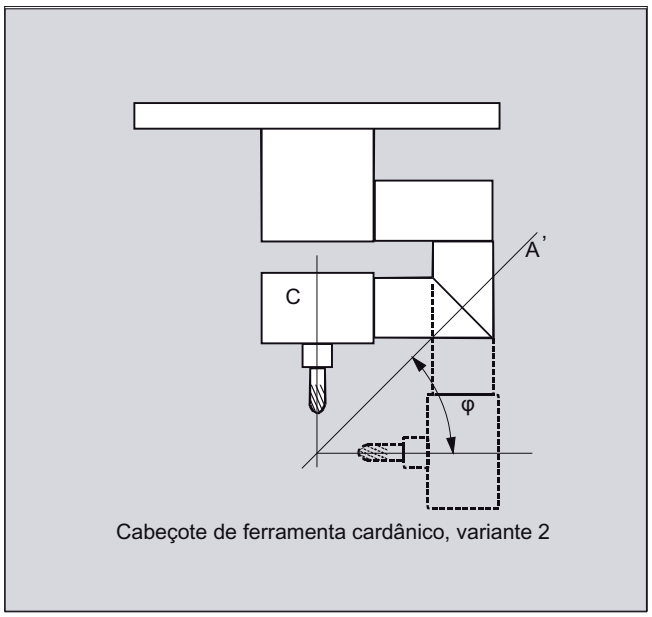


Cabeçote de ferramenta cardânico, variante 1

Nos exemplos aqui apresentados temos as disposições no exemplo da cinemática de máquina CA com o cabeçote de ferramenta cardânico!

Fabricante da máquina

A seqüência dos eixos de orientação e o sentido de orientação da ferramenta pode ser ajustado em função da cinemática da máquina através dos dados de máquina.



Cabeçote de ferramenta cardânico, variante 2

Neste exemplo o A' encontra-se a um ângulo φ em relação ao eixo X

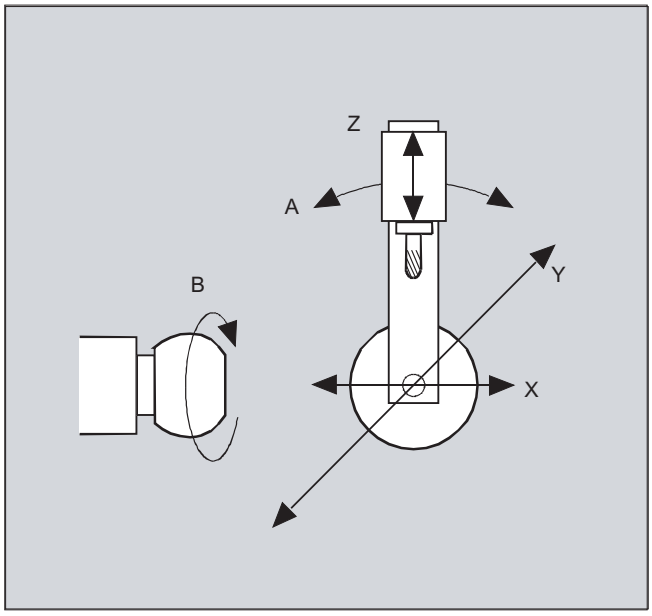
Normalmente são aplicadas as seguintes possíveis relações:

A' encontra-se a um ângulo φ em relação ao	Eixo X
B' encontra-se a um ângulo φ em relação ao	Eixo Y
C' encontra-se a um ângulo φ em relação ao	Eixo Z

O ângulo φ pode ser configurado na faixa de 0° até +89° através de dados de máquina.

Com eixo linear giratório

Aqui trata-se de uma disposição com peça de trabalho em movimento e ferramenta em movimento. A cinemática é formada a partir de três eixos lineares (X, Y, Z) e dois eixos de rotação dispostos ortogonalmente. Por exemplo, o primeiro eixo rotativo é movimentado por uma unidade de avanço cruzada com dois eixos lineares, a ferramenta está paralela ao terceiro eixo linear. O segundo eixo de rotação gira a peça de trabalho. O terceiro eixo linear (eixo que gira) está no plano da unidade de avanço cruzada.



A seqüência dos eixos rotativos e o sentido de orientação da ferramenta pode ser ajustado em função da cinemática da máquina através dos dados de máquina.

São aplicadas as seguintes possíveis relações:

Eixos:	Seqüências de eixos:
1. Eixo rotativo	A A B B C C
2. Eixo rotativo	B C A C A B
Eixo linear girado	Z Y Z X Y X

Para mais explicações sobre a configuração de seqüências de eixos para o sentido de orientação da ferramenta, veja

Literatura: /FB3/ Manual de funções especiais; Transformações de 3 a 5 eixos (F2), Capítulo Cabeçote de fresar cardânico, "Parametrização".

6.2.2 Transformação de três, quatro e cinco eixos (TRAORI)

Função

O usuário pode configurar dois ou três eixos translatórios e um eixo rotatório. As transformações partem do princípio de que o eixo rotatório esteja em posição ortogonal ao plano de orientação.

A orientação da ferramenta somente é possível no plano que estiver perpendicular ao eixo rotatório. A transformação dá suporte aos tipos de máquina com ferramenta articulada e peça de trabalho móvel.

A projeção e programação de transformações de três e quatro eixos é realizada de forma análoga às transformações de cinco eixos.

Literatura:
Manual de funções especiais; Transformações múltiplas (F2)

Sintaxe

```
TRAORI(<n>)
TRAORI(<n>,<X>,<Y>,<Z>,<A>,<B>)
TRAFOOF
```

Significado

`TRAORI`:	Ativa a primeira transformação de orientação especificada
`TRAORI(<n>)`:	Ativa a transformação de orientação especificada com n
`<n>`:	Número da transformação Valor: 1 ou 2 Exemplo: TRAORI(1) ativa a transformação de orientação 1
`<X>,<Y>,<Z>`:	Componente do vetor de orientação em que a ferramenta aponta.
`<A>,<B>`:	Offset programável para os eixos rotativos
`TRAFOOF`:	Desativação da transformação

Orientação da ferramenta

Em função do sentido de orientação selecionado da ferramenta, no programa NC o plano de trabalho ativo (G17, G18, G19) deve ser ajustado de modo que a correção do comprimento da ferramenta atue no sentido da orientação da ferramenta.

Indicação

Após a ativação da transformação as posições (Z, Y, Z) sempre estão relativas à ponta da ferramenta. Alteração das posições dos eixos rotativos envolvidos na transformação que gera os tais movimentos de compensação dos demais eixos da máquina, de modo que a posição da ponta da ferramenta permaneça inalterada.

A transformação da orientação sempre está alinhada da ponta da ferramenta para a fixação da ferramenta.

Offset para eixos de orientação

Na ativação da transformação da orientação se pode programar diretamente um deslocamento adicional para eixos de orientação.

Os parâmetros podem ser omitidos se na programação for mantida a sequência correta.

Exemplo:

`TRAORI(, , , ,A,B)` ; se apenas um único Offset deve ser especificado

Como alternativa à programação direta, o Offset adicional para eixos de orientação também pode ser transferido automaticamente do atual deslocamento do ponto zero ativo. A transferência é configurada através de dados de máquina.

Exemplos

```
TRAORI(1,0,0,1)          ; A orientação básica da ferramenta aponta para o sentido Z

TRAORI(1,0,1,0)          ; A orientação básica da ferramenta aponta para o sentido Z

TRAORI(1,0,1,1)          ; A orientação básica da ferramenta aponta para o sentido
                           Y/Z (corresponde a posição -45°)
```

6.2.3 Variantes da programação da orientação e posição básica (ORIRESET)

Programação de orientação da orientação de ferramenta com TRAORI

Em conjunto com uma transformação de orientação programável TRAORI, além dos eixos lineares X, Y e Z, também se pode programar posições de eixo através dos indicadores de eixo rotativo A.., B..., C... ou eixos virtuais com ângulos ou componentes de vetor. Para eixos de orientação e eixos de máquina são possíveis diversos tipos de interpolação. Independentemente de quais polinômios de orientação PO[ângulo] e polinômios de eixo PO[eixo] estão ativos no momento, se pode programar diversos tipos diferentes de polinômio como G1, G2, G3, CIP ou POLY.

A alteração da orientação da ferramenta também pode ser programada através de vetores de orientação. Neste caso, a orientação final de cada bloco pode ser realizada através de programação direta do vetor ou através da programação das posições de eixo rotativo.

Indicação

Variantes da programação de orientação em transformações de três a cinco eixos

Na transformação de três a cinco eixos as variantes

1. A, B, C Indicação direta das posições de eixo da máquina
2. A2, B2, C2 Programação de ângulo de eixos virtuais através de ângulos eulerianos ou RPY
3. A3 ,B3, C3 Indicação de componentes de vetor
4. LEAD, TILT Indicação do ângulo de avanço e do ângulo lateral sobre a trajetória e superfície
5. A4, B4, C4 e A5, B5, C5 Vetor normal de superfície no início do bloco e no fim do bloco
6. A6, B6, C6 e A7, B7, C7 Interpolação do vetor de orientação sobre uma superfície periférica cônica
7. A8, B8, C8 Reorientação da ferramenta, sentido e distância de curso do movimento de retração

se excluem uma às outras.

Valores programados de forma misturada também são eliminados com mensagens de alarme.

Posição básica da orientação da ferramenta ORIRESET

Através da programação do ORIRESET(A, B, C) os eixos de orientação são deslocados de forma linear e sincronizada de sua posição momentânea até a posição básica especificada.

Se não for programada nenhuma posição básica para um eixo, então será utilizada a posição definida no dado de máquina correspondente $MC_TRAFO5_ROT_AX_OFFSET_1/2. Neste caso não serão considerados eventuais Frames ativos dos eixos rotativos.

Indicação

Somente quando uma transformação de orientação estiver ativa com TRAORI(...) é que se pode programar uma posição básica da orientação da ferramenta com ORIRESET(...), independente de cinemática e sem alarme 14101.

Exemplos

```
1. Exemplo para cinemática de máquina CA (nomes de eixos de canal C, A)
ORIRESET(90, 45)       ;C em 90 graus, A em 45 graus
ORIRESET(, 30)         ;C em $MC_TRAFO5_ROT_AX_OFFSET_1/2[0], A em 30 graus
ORIRESET( )            ;C em $MC_TRAFO5_ROT_AX_OFFSET_1/2[0],
                       ;A em $MC_TRAFO5_ROT_AX_OFFSET_1/2[1]
2. Exemplo para cinemática de máquina CAC (nomes de eixos de canal C, A, B)
ORIRESET(90, 45, 90)   ;C em 90 graus, A em 45 graus, B em 90 graus
ORIRESET( )            ;C em $MC_TRAFO5_ROT_AX_OFFSET_1/2[0],
                       ;A em $MC_TRAFO5_ROT_AX_OFFSET_1/2[1],
                       ;B em $MC_TRAFO5_ROT_AX_OFFSET_1/2[2]
```

Programação das rotações LEAD, TILT e THETA

As rotações da orientação da ferramenta são programadas na transformação de três a cinco eixos como ângulo de avanço LEAD e o ângulo lateral TILT.

Em uma transformação com terceiro eixo rotativo são permitidas programações adicionais com C2 (rotações do vetor de orientação) tanto na orientação com componentes de vetor como com indicação do ângulo LEAD e TILT.

Com um terceiro eixo rotativo adicional se pode programar a rotação da ferramenta em torno de si mesma através do ângulo de rotação THETA.

6.2.4 Programação da orientação da ferramenta (A..., B..., C..., LEAD, TILT)

Função

Para a programação da orientação da ferramenta existem as seguintes opções:

1. Programação direta do movimento dos eixos rotativos. A alteração da orientação sempre é realizada no sistema de coordenadas básico ou no sistema de coordenadas da máquina. Os eixos de orientação são movimentados como eixos sincronizados.
2. Programação em ângulos eulerianos ou RPY conforme definição angular através do `A2, B2, C2`.
3. Programação do vetor de direção através do `A3, B3, C3`. O vetor de direção aponta para da ponta da ferramenta no sentido da fixação da ferramenta.
4. Programação do vetor normal de superfície no início do bloco com `A4, B4, C4` e no fim do bloco com `A5, B5, C5` (fresamento de topo).
5. Programação através de ângulo de avanço `LEAD` e ângulo lateral `TILT`
6. Para programação do eixo de rotação do cone como vetor normalizado através do `A6, B6, C6` ou da orientação intermediária sobre a superfície periférica cônica através do `A7, B7, C7,`
   veja o capítulo "Programação da orientação ao longo da superfície periférica cônica (ORIPLANE, ORICONxx)".
7. Para programação da reorientação, direção e distância de curso da ferramenta durante o movimento de retração através do `A8, B8, C8,`
   veja o capítulo "Suavização do decurso de orientação (ORIPATHS A8=, B8=, C8=)"

Indicação

Em todos casos a programação da orientação somente é permitida se uma transformação de orientação estiver ativada.

Vantagem: Estes programas podem ser transferidos para qualquer cinemática de máquina.

Definição da orientação da ferramenta através de código G

Indicação

Fabricante da máquina

Através do dado da máquina se pode comutar entre ângulos eulerianos ou ângulos RPY. De acordo com as respectivas configurações de dados de máquina uma comutação é possível tanto dependente como independente do código G ativo do grupo 50. Estão disponíveis as seguintes opções de configuração:

1. Se os dois dados de máquina estiverem ajustados com valor zero para a definição dos eixos de orientação e definição do ângulo de orientação através de código G:
   Os ângulos programados com `A2, B2, C2` podem ser interpretados como ângulos eulerianos ou como ângulos RPY **em função do dado de máquina** da definição de ângulos da programação de orientação.
2. Se o dado de máquina para a definição dos eixos de orientação através de código G estiver ajustado com valor um, a comutação é realizada **em função** do código G ativo do grupo 50:
   Os ângulos programados com `A2, B2, C2` são interpretados de acordo com um dos códigos G ativos `ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2, ORIAXPOS` e `ORIPY2` . Os valores programados com os eixos de orientação também são interpretados como ângulo de orientação de acordo com o código G ativo do grupo 50.
3. Se o dado de máquina para a definição do ângulo de orientação através de código G for ajustado com valor um e o dado de máquina para a definição dos eixos de orientação através de código G for ajustado com valor zero, a comutação é realizada de forma **independente** do código G ativo do grupo 50:
   Os ângulos programados com `A2, B2, C2` são interpretados de acordo com um dos códigos G ativos `ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2 ORIAXPOS` e `ORIPY2` . Os valores programados com os eixos de orientação sempre são interpretados como posições de eixo rotativo independentemente do código G ativo do grupo 50.

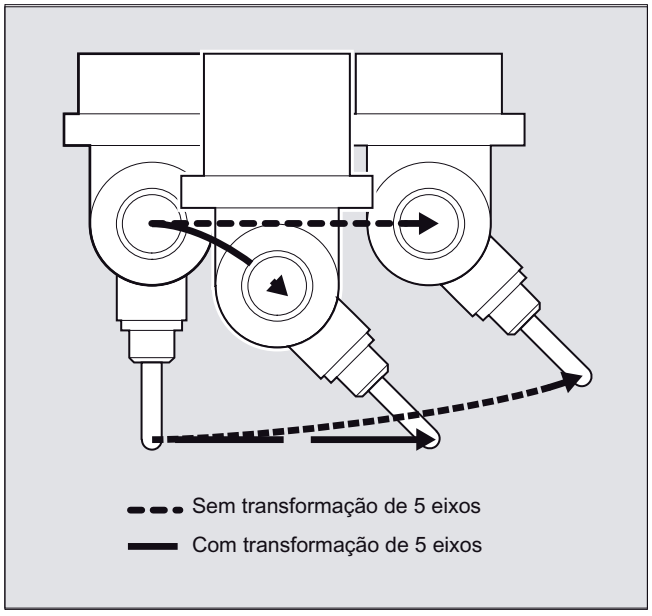
Programação

`G1 X Y Z A B C`	Programação do movimento dos eixos rotativos
`G1 X Y Z A2= B2= C2=`	Programação em ângulos eulerianos
`G1 X Y Z A3== B3== C3==`	Programação do vetor de direção
`G1 X Y Z A4== B4== C4==`	Programação do vetor normal de superfície no início do bloco
`G1 X Y Z A5== B5== C5==`	Programação do vetor normal de superfície no fim do bloco
`LEAD=`	Ângulo de avanço para a programação da orientação da ferramenta
`TILT=`	Ângulo lateral para a programação da orientação da ferramenta

Parâmetro

G....	Indicação do tipo de movimento dos eixos rotativos
X Y Z	Indicação dos eixos lineares
A B C	Indicação das posições de eixo da máquina dos eixos rotativos
A2 B2 C2	Programação de ângulos (eulerianos ou RPY) de eixos virtuais ou eixos de orientação
A3 B3 C3	Indicação dos componentes do vetor de direção
A4 B4 C4	Indicação, por exemplo no fresamento de topo, dos componentes do vetor normal de superfície no início do bloco
A5 B5 C5	Indicação, por exemplo no fresamento de topo, dos componentes do vetor normal de superfície no fim do bloco
LEAD	Ângulo relativo ao vetor normal de superfície, no plano fixado pela tangente de percurso e vetor normal de superfície
TILT	Ângulo no plano, perpendicular à tangente de percurso relativa ao vetor normal de superfície

Exemplo de comparação de sem e com transformação de 5 eixos



Descrição

Normalmente os programas de 5 eixos são gerados a partir de sistemas CAD/CAM e não são especificados no comando. Por isso que as seguintes explicações estão voltadas principalmente aos programadores de pós-processadores.

O tipo de programação de orientação é definida no código G do grupo 50:

`ORIEULER` através de ângulo euleriano
`ORIRPY` através de ângulo RPY (seqüência de rotação ZYX)
`ORIVIRT1` através de eixos virtuais de orientação (definição 1)
`ORIVIRT2` através de eixos virtuais de orientação (definição 2)
`ORIAXPOS` através de eixos virtuais de orientação com posições de eixo rotativo
`ORIPY2` através de ângulo RPY (seqüência de rotação XYZ)

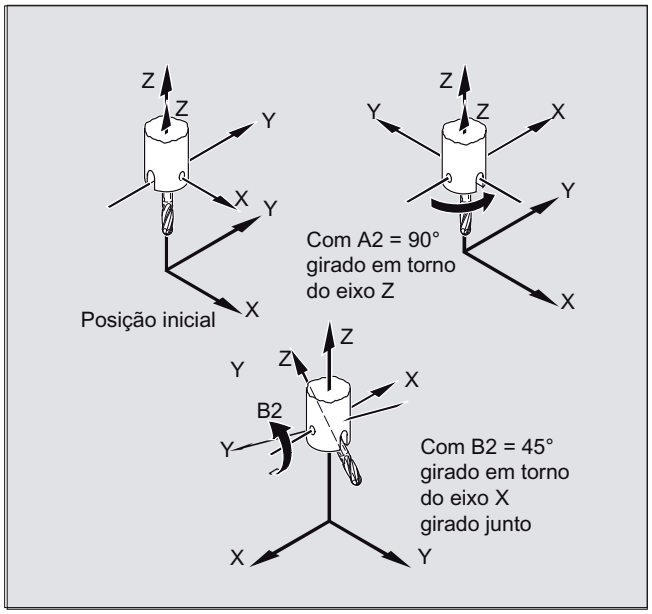
Fabricante da máquina

Através de dados de máquina podem ser definidos diversas variantes pelo fabricante da máquina. Observe as instruções do fabricante da máquina.

Programação em ângulos eulerianos ORIEULER

Os valores programados com `A2, B2, C2` na programação da orientação são interpretados como ângulo euleriano (em graus).

O vetor de orientação é formado quando um vetor é girado no sentido Z primeiramente com `A2` em torno do eixo Z, depois com `B2` em torno do novo eixo X e por último com `C2` em torno do novo eixo Z.



Neste caso o valor de `C2` (rotação em torno do novo eixo Z) não tem significado e não precisa ser programado.

Programação em ângulos RPY ORIRPY

Os valores programados com `A2, B2, C2` na programação da orientação são interpretados como ângulo RPY (em graus).

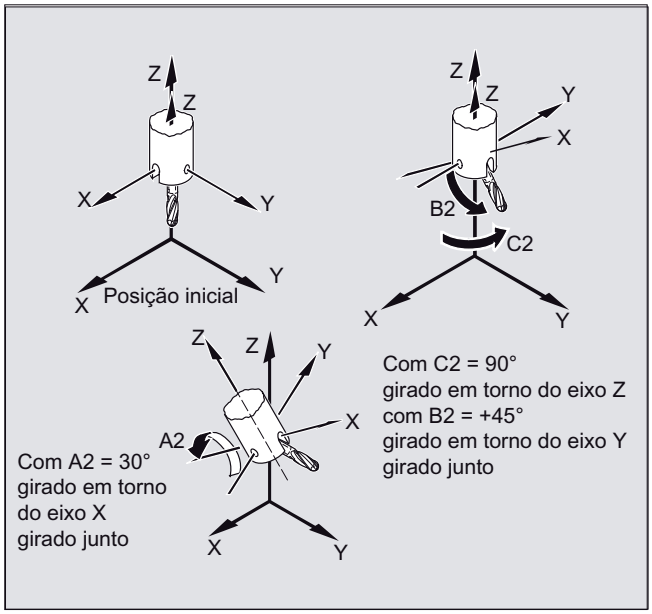
Indicação

Ao contrário da programação com ângulos eulerianos, aqui os três valores têm influência sobre o vetor da orientação.

Fabricante da máquina

Na definição de ângulos com ângulo de orientação através de ângulos RPY se aplica $MC_ORI_DEF_WITH_G_CODE = 0 para os eixos de orientação

O vetor de orientação é formado quando um vetor é girado no sentido Z primeiramente com C2 em torno do eixo Z, depois com B2 em torno do novo eixo Y e por último com A2 em torno do novo eixo X.

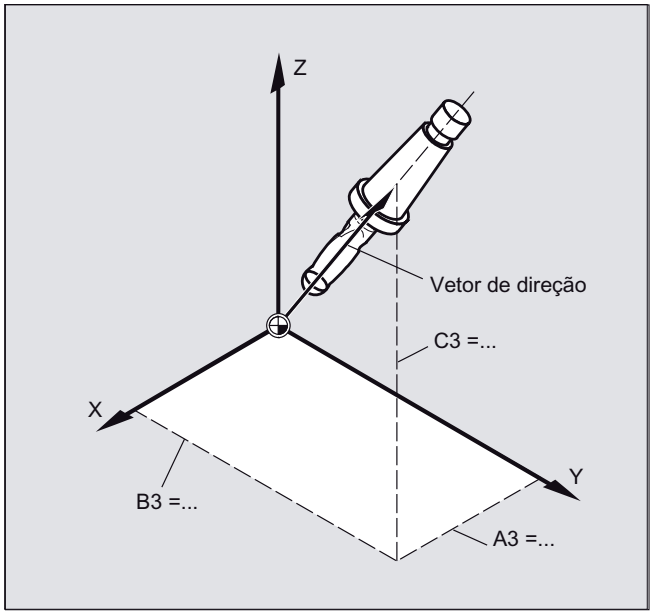


Se o dado de máquina para definição dos eixos de orientação através de código G estiver ajustado com $MC_ORI_DEF_WITH_G_CODE = 1, então se aplica:
O vetor de orientação é formado quando um vetor é girado no sentido Z primeiramente com A2 em torno do eixo Z, depois com B2 em torno do novo eixo Y e por último com C2 em torno do novo eixo X.

Programação do vetor de direção

Os componentes do vetor de direção são programados com `A3, B3, C3`. O vetor aponta para a direção da fixação da ferramenta; o comprimento do vetor é irrelevante neste caso.

Os componentes não programados são definidos igual a zero.



Programação da orientação da ferramenta com LEAD= e TILT=

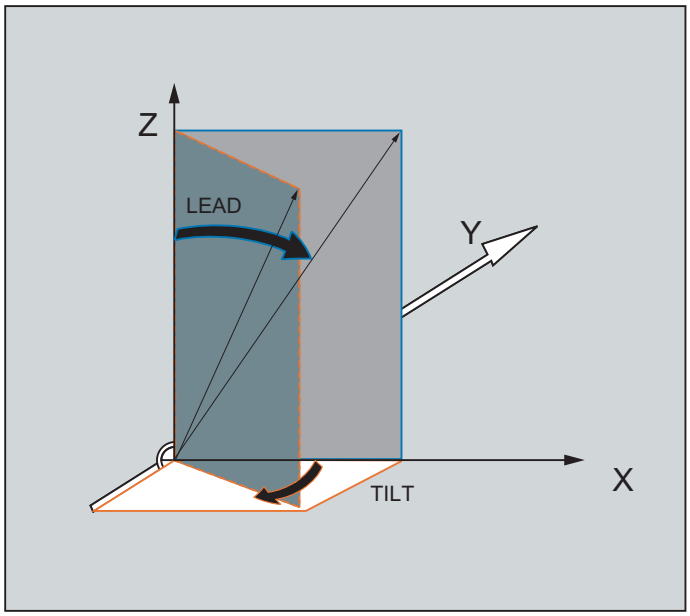
A orientação da ferramenta resultante é obtida a partir do(a):

- Tangente de percurso
- Vetor normal de superfície
  no início do bloco `A4, B4, C4` e no fim do bloco `A5, B6, C5`
- Ângulo de avanço `LEAD`
  no plano fixado pela tangente de percurso e pelo vetor normal de superfície
- Ângulo lateral `TILT` no fim do bloco
  perpendicular à tangente de percurso e relativo ao vetor normal de superfície

Comportamento nos cantos internos (para WZK 3D)

Se o bloco for truncado em um canto interno, a orientação da ferramenta resultante também será alcançada no fim do bloco.

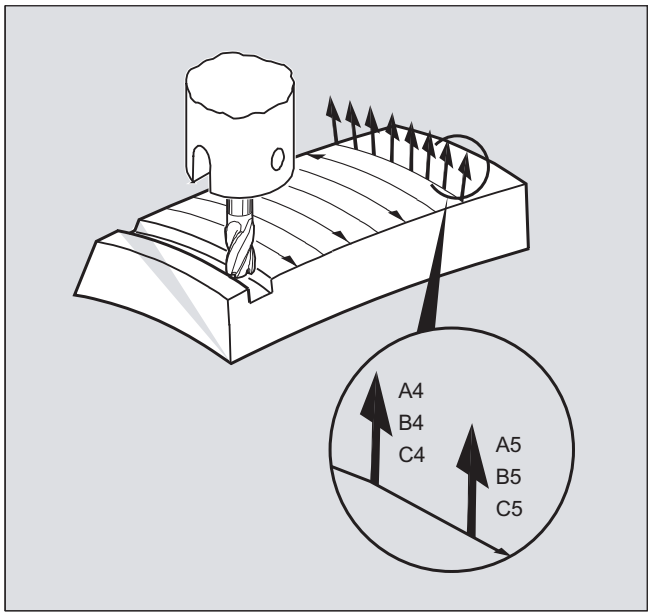
Definição da orientação da ferramenta com LEAD= e TILT=



6.2.5 Fresamento de topo (fresamento 3D A4, B4, C4, A5, B5, C5)

Função

O fresamento de topo serve para usinagem de qualquer superfície curvada.



Para este tipo de fresamento 3D precisamos da descrição das trajetórias 3D sobre a superfície da peça de trabalho.

Normalmente no CAM os cálculos são executados considerando-se a forma e dimensões da ferramenta. Depois os blocos NC calculados são inseridos no comando numérico através de pós-processamento.

Programação da curvatura de trajetória

Descrição das superfícies

A descrição da curvatura de trajetória é realizada através de vetores normais de superfície com os seguintes componentes:

`A4, B4, C4` Vetor inicial no início do bloco

`A5, B5, C5` Vetor final no fim do bloco

Se em um bloco existir apenas o vetor inicial, o vetor normal de superfície permanecerá constante por todo bloco. Se em um bloco existir apenas o vetor final, então será realizada a interpolação de grande circunferência a partir do valor final do bloco anterior até o valor final programado.

Se foram programados vetor inicial e final, então também se realiza a interpolação de grande circunferência entre os dois sentidos. Com isso são obtidos percursos suaves e constantes.

No ajuste básico os vetores normais de superfície apontam para o sentido Z, independentemente do plano ativo G17 até G19.

O comprimento de um vetor é irrelevante.

Os componentes de vetor não programados são definidos como zero.

Com o ORIWKS, veja o capítulo "Relação dos eixos de orientação (ORIWKS, ORIMKS)", os vetores normais de superfície são relativos ao Frame ativo e são girados junto com o Frame.

Fabricante da máquina

O vetor normal de superfície deve estar perpendicular em relação à tangente do percurso, dentro de um valor limite ajustável via dado da máquina, caso contrário será emitido um alarme.

6.2.6 Relação dos eixos de orientação (ORIWKS, ORIMKS)

Função

Na programação da orientação no sistema de coordenadas da peça através

- do ângulo euleriano e ângulo RPY ou
- vetor de orientação

se pode ajustar o decurso do movimento de rotação através do ORIMKS/ORIWKS.

Indicação

Fabricante da máquina

O tipo de interpolação para a orientação é definido com o dado de máquina:

MD21104 $MC_ORI_IPO_WITH_G_CODE

=FALSE: A referência são as funções G ORIWKS e ORIMKS

=TRUE: A referência são as funções G do 51º grupo (ORIAXES, ORIVECT, ORIPLANE, ...)

Sintaxe

```
ORIMKS=...
ORIWKS=...
```

Significado

ORIMKS	Rotação no sistema de coordenadas da máquina
ORIWKS	Rotação no sistema de coordenadas da peça de trabalho

Indicação

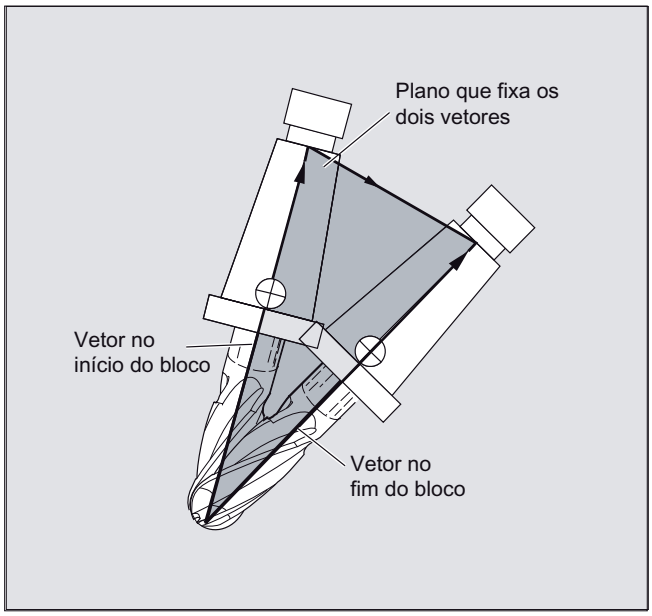
ORIWKS é ajuste básico. No caso de um programa de cinco eixos onde ainda não se sabe em qual máquina ele deverá ser executado, sempre se deve selecionar ORIWKS. Quais movimentos que a máquina realmente executa, depende da cinemática da máquina.

Por exemplo, com ORIMKS podem ser programados os movimentos reais da máquina para evitar colisões com dispositivos ou similares.

Descrição

Com o ORIMKS o movimento executado pela ferramenta **depende** da cinemática da máquina. Para a alteração da orientação com a ponta da ferramenta fixa no espaço, se realiza a interpolação de modo linear entre as posições dos eixos rotativos.

Com o ORIWKS o movimento executado pela ferramenta **não depende** da cinemática da máquina. Para a alteração da orientação com a ponta da ferramenta fixa no espaço, a ferramenta movimenta-se no plano formado pelos vetores inicial e final.



Posições singulares

Indicação

ORIWKS

Movimentos de orientação na área da posição singular da máquina de cinco eixos requerem grandes movimentos dos eixos da máquina. (Por exemplo, em um cabeçote rotativo tendo C como eixo de rotação e A como eixo de orientação, todas posições com A=0 são singulares.)

Fabricante da máquina

Para não sobrecarregar os eixos da máquina, o controle de velocidade reduz bastante a velocidade de percurso próximo às posições singulares.

Com os dados da máquina

$MC_TRAFO5_NON_POLE_LIMIT

$MC_TRAFO5_POLE_LIMIT

a transformação pode ser parametrizada de modo que os movimentos de orientação na proximidade do pólo sejam colocados pelo pólo e com isso seja possível realizar uma usinagem rápida.

As posições singulares são tratadas apenas com o MD $MC_TRAFO5_POLE_LIMIT .

Literatura:
/FB3/ Manual de funções especiais; Transformação de 3 a 5 eixos (F2),
Capítulo "Posições singulares e seu tratamento".

6.2.7 Programação dos eixos de orientação (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2)

Função

A função de eixos de orientação descreve a orientação da ferramenta no espaço e é alcançada através da programação dos Offsets para eixos rotativos. Um terceiro grau de liberdade pode ser obtido através da rotação adicional da ferramenta em torno de si mesma. Esta orientação de ferramenta é realizada em qualquer ponto no espaço através de um terceiro eixo rotativo e requer a transformação de seis eixos. A rotação própria da ferramenta em torno de si mesma é definida em função do tipo de interpolação dos vetores de rotação com o ângulo de rotação THETA, veja o capítulo "Rotações da orientação da ferramenta (ORIROTA/TR/TT, ORIROTC, THETA)".

Programação

Os eixos de orientação são programados através dos identificadores de eixo A2, B2 e C2.

`N... ORIAXES` ou `ORIVECT` `N... G1 X Y Z A B C`	Interpolação linear ou de grande circunferência
ou	ou
`N... ORIPLANE`	Interpolação de orientação do plano
ou	ou
`N... ORIEULER` ou `ORIRPY e ORIRPY2` `N... G1 X Y Z A2= B2= C2=`	Ângulo de orientação através de ângulo euleriano/RPY
ou	ou
`N... ORIVIRT1` ou `ORIVIRT2` `N... G1 X Y Z A3= B3= C3=`	Programação de ângulo dos eixos virtuais ou eixos virtuais de orientação da definição 1 ou 2 da programação de vetor de direção

Para alterações de orientação ao longo de uma superfície presente no espaço podem ser programados outros Offsets de eixo rotativo dos eixos de orientação, veja o capítulo "Programação da orientação ao longo da superfície periférica cônica (ORIPLANE, ORICONxx).

Parâmetro

ORIAXES	Interpolação linear dos eixos da máquina ou dos eixos de orientação
ORIVECT	Interpolação de grande circunferência (idêntico ao ORIPLANE)
ORIMKS	Rotação no sistema de coordenadas da máquina
ORIWKS	Rotação no sistema de coordenadas da peça de trabalho
	Veja a descrição no cap. Rotações da orientação da ferramenta
A= B= C=	Programação da posição de eixo da máquina
ORIEULER	Programação da orientação através de ângulo euleriano
ORIRPY	Programação da orientação através de ângulo RPY. A seqüência de rotação é XYZ, onde se aplica: A2 é o ângulo de rotação em torno do X B2 é o ângulo de rotação em torno do Y C2 é o ângulo de rotação em torno do Z
ORIRPY2	Programação da orientação através de ângulo RPY. A seqüência de rotação é ZYX, onde se aplica: A2 é o ângulo de rotação em torno do Z B2 é o ângulo de rotação em torno do Y C2 é o ângulo de rotação em torno do X
A2= B2= C2=	Programação de ângulo dos eixos virtuais
ORIVIRT1 ORIVIRT2	Programação da orientação através de eixos virtuais de orientação
	(definição 1), definição conforme MD $MC_ORIAX_TURN_TAB_1
	(definição 2), definição conforme MD $MC_ORIAX_TURN_TAB_2
A3= B3= C3=	Programação do vetor de direção do eixo de sentido

Descrição

Fabricante da máquina

Com MD $MC_ORI_DEF_WITH_G_CODE se define como os ângulos A2, B2, C2 programados serão definidos:

A definição é realizada conforme o MD $MC_ORIENTATION_IS_EULER (Standard) ou a definição é realizada conforme o grupo 50 de códigos G (ORIEULER, ORIRPY, ORIVIRT1, ORIVIRT2).

Com MD $MC_ORI_IPO_WITH_G_CODE se define qual tipo de interpolação está ativa: ORIWKS/ORIMKS ou ORIAXES/ORIVECT.

Modo de operação JOG

Os ângulos de orientação sempre são interpolados linearmente neste modo de operação. Com o deslocamento contínuo e incremental através das teclas de deslocamento somente pode ser movimentado um eixo de orientação. Através das manivelas eletrônicas os eixos de orientação podem ser movimentados simultaneamente.

Para o movimento manual dos eixos de orientação, o interruptor de correção (override) de avanço ou o interruptor de correção de avanço rápido tem efeito na correção do avanço rápido.

Com os seguintes dados da máquina é possível realizar uma especificação separada de velocidade:

$MC_JOG_VELO_RAPID_GEO

$MC_JOG_VELO_GEO

$MC_JOG_VELO_RAPID_ORI

$MC_JOG_VELO_ORI

Indicação

SINUMERIK 840D com "Pacote de transformação Handling"

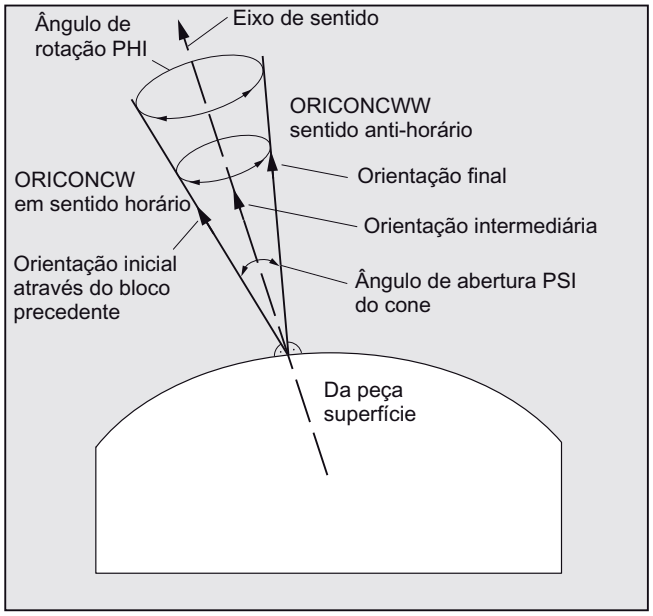
Com a função "Deslocamento manual cartesiano", em modo JOG, se pode ajustar separadamente a translação dos eixos geométricos no sistema de referência MCS, WCS e TCS.

Literatura:
/FB2/ Manual de funções ampliadas; Transformação cinemática (M1)

6.2.8 Programação da orientação ao longo de uma superfície periférica cônica (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO)

Função

Com a orientação ampliada é possível executar alterações da orientação ao longo de uma superfície periférica cônica no espaço. A interpolação do vetor de orientação sobre uma superfície periférica cônica é realizada com os comandos modais ORICONxx. Para a interpolação em um plano se pode programar a orientação final com ORIPLANE. Geralmente a orientação inicial é definida através dos blocos anteriores.



Programação

A orientação final é definida através da indicação da programação em ângulos eulerianos ou RPY com A2, B2, C2 ou através da programação das posições de eixo rotativo com A, B, C. Para os eixos de orientação ao longo da superfície periférica cônica são necessárias outras informações de programação:

- Eixo de rotação do cone como vetor com `A6, B6, C6`
- Ângulo de abertura `PSI` com o identificador `NUT`
- Orientação intermediária com superfície cônica com `A7, B7, C7`

Indicação

Programação do vetor de direção A6, B6, C6 par ao eixo de rotação do cone

A programação de uma orientação final não é extremamente necessária. Se não for especificada nenhuma orientação final, então se interpola uma superfície cônica inteira com 360 graus.

Programação do ângulo de abertura do cone com NUT=ângulo

A indicação de uma orientação final é extremamente necessária.

Uma superfície cônica completa com 360 graus não pode ser interpolada dessa forma.

Programação da orientação intermediária A7, B7, C7 na superfície cônica

A indicação de uma orientação final é extremamente necessária. A alteração da orientação e o sentido de giro são claramente definidos através dos três vetores de orientação inicial, final e intermediário. Neste caso todos os três vetores devem ser diferentes entre si. Se a orientação intermediária programada for paralela à orientação inicial ou final, então será realizada uma interpolação linear de grande circunferência da orientação no plano que foi fixado pelos vetores inicial e final.

Interpolação ampliada de orientação sobre uma superfície periférica cônica

```
N... ORICONCW ou ORICONCCW
N... A6= B6= C6= A3= B3= C3=
ou
N... ORICONTO
N... G1 X Y Z A6= B6= C6=
ou
N... ORICONIO
N... G1 X Y Z A7= B7= C7=
N... PO[PHI]=(a2, a3, a4, a5)
N... PO[PSI]=(b2, b3, b4, b5)
```

Interpolação sobre uma superfície cônica com

vetor de direção em sentido horário/anti-horário do cone e orientação final ou

transição tangencial e
indicação da orientação final

ou
indicação da orientação final e uma orientação intermediária na superfície cônica com

polinômios para ângulo de rotação e polinômios para ângulo de abertura

Parâmetros

ORIPLANE	Interpolação no plano (interpolação de grande circunferência)
ORICONCW	Interpolação sobre uma superfície periférica cônica no sentido horário
ORICONCCW	Interpolação sobre uma superfície periférica cônica no sentido anti-horário
ORICONTO	Interpolação sobre uma superfície periférica cônica de transição tangencial
A6= B6= C6=	Programação do eixo de rotação do cone (vetor normalizado)
NUT=ângulo	Ângulo de abertura do cone em graus
NUT=+179	Ângulo de deslocamento menor ou igual a 180 graus
NUT=-181	Ângulo de deslocamento maior ou igual a 180 graus
ORICONIO	Interpolação sobre uma superfície periférica cônica
A7= B7= C7=	Orientação intermediária (programação como vetor normalizado)
PHI	Ângulo de rotação da orientação em torno do eixo de sentido do cone
PSI	Ângulo de abertura do cone
possíveis polinômios PO[PHI]=(a2, a3, a4, a5) PO[PSI]=(b2, b3, b4, b5)	Além dos respectivos ângulos também se pode programar polinômios de até 5º grau

Exemplo de diferentes alterações de orientação

```
…
N10 G1 X0 Y0 F5000
N20 TRAORI(1)                        ;  Transformação de orientação ativada.
N30 ORIVECT                          ;  Interpolação da orientação de ferramenta como
                                        vetor.
…                                    ;  Orientação de ferramenta no plano.
N40 ORIPLANE                         ;  Seleção de interpolação de grande
                                        circunferência.
N50 A3=0 B3=0 C3=1
N60 A3=0 B3=1 C3=1                   ;  Orientação no plano Y/Z girada em 45 graus; a
                                        orientação é alcançada no fim do bloco
                                        (0,1/√2,1/√2).
…
N70 ORICONCW                         ;  Programação da orientação na superfície
                                        cônica:
N80 A6=0 B6=0 C6=1 A3=0 B3=0 C3=1    ;  O vetor de orientação é interpolado em
                                        sentido horário sobre uma superfície cônica
                                        com a direção (0,0,1) até a orientação
                                        (1/√2,0,1/√2), sendo que o ângulo de rotação
                                        é de 270 graus.
N90 A6=0 B6=0 C6=1                   ;  A orientação de ferramenta percorre um giro
                                        inteiro sobre a mesma superfície cônica.
```

Descrição

Quando devem ser descritas alterações de orientação em uma superfície periférica cônica qualquer no espaço, então o vetor com que a orientação de ferramenta deve ser girado deve ser conhecido. Além disso, se deve especificar a orientação inicial e final. A orientação inicial resulta do bloco anterior e a orientação final deve ser programada ou definida por outras condições.

A programação no plano ORIPLANE corresponde ao ORIVECT

A programação da interpolação de grande circunferência junto com polinômios angulares corresponde à interpolação linear e de polinômios de contornos. A orientação de ferramenta é interpolada neste plano, que foi fixado pela orientação inicial e orientação final. Se também forem programados polinômios, então o vetor de orientação também pode ser excluído do plano.

Programação de círculos em um plano G2/G3, CIP e CT

A orientação ampliada corresponde à interpolação de círculos em um plano. Sobre as respectivas possibilidades de programação de círculos com indicação de centro ou indicação de raio como G2/G3, círculo através de ponto intermediário CIP e círculos tangenciais CT, veja

Literatura: Manual de programação Fundamentos, "Programar comandos de curso".

Programação da orientação

Interpolação do vetor de orientação sobre uma superfície periférica cônica ORICONxx

Para a interpolação de orientações sobre uma superfície periférica cônica podem ser selecionados quatro tipos diferentes de interpolação do grupo 51 de códigos G:

1. Interpolação sobre uma superfície cônica em sentido horário `ORICONCW` com indicação da orientação final e do sentido do cone ou do ângulo de abertura. O vetor de direção é programado com os identificadores `A6, B6, C6` e o ângulo de abertura do cone é programado com o identificador NUT= na faixa de valores de 0 a 180 graus.
2. Interpolação sobre uma superfície cônica em sentido anti-horário `ORICONCWW` com indicação da orientação final e do sentido do cone ou do ângulo de abertura. O vetor de direção é programado com os identificadores `A6, B6, C6` e o ângulo de abertura do cone é programado com o identificador NUT= na faixa de valores de 0 a 180 graus.
3. Interpolação sobre uma superfície cônica `ORICONIO` com indicação da orientação final e uma orientação intermediária, que é programada com os identificadores `A7, B7, C7`.
4. Interpolação sobre uma superfície cônica `ORICONTO` com transição tangencial e indicação da orientação final. O vetor de direção é programado com os identificadores `A6, B6, C6`.

6.2.9 Especificação de orientação de dois pontos de contato (ORICURVE, PO[XH]=, PO[YH]=, PO[ZH]=)

Função

Programação da alteração de orientação através da segunda curva espacial ORICURVE

Uma outra possibilidade de programação de alterações de orientação é que além da ponta da ferramenta também se pode, ao longo de uma curva espacial, programar o movimento de um segundo ponto de contato da ferramenta com `ORICURVE`. Com isso se pode definir claramente as alterações de orientação da ferramenta, como na programação do próprio vetor de ferramenta.

Fabricante da máquina

Observe as instruções do fabricante da máquina sobre os identificadores de eixo ajustados em dados de máquina ao programar a 2ª trajetória de orientação da ferramenta.

Programação

Neste tipo de interpolação se pode programar pontos com `G1` e polinômios com `POLY` para as duas curvas espaciais. Círculos e evolventes não são permitidos. Adicionalmente pode ser ativada uma interpolação de Spline com `BSPLINE` e a função "Agrupamento de blocos Spline curtos".

Literatura:
/FB1/ Manual de funções básicas; Modo de controle da trajetória, Parada exata, Look Ahead (B1), Capítulo: Agrupamento de blocos Spline curtos

Não são permitidos outros tipos de Spline `ASPLINE` e `CSPLINE`, assim como a ativação de um compressor com `COMPON`, `COMPCURV` ou `COMPCAD`.

O movimento de dois pontos de contato da ferramenta pode ser especificado na programação dos polinômios de orientação para coordenadas de até 5º grau.

Interpolação de orientação ampliada com curva espacial adicional e polinômios para coordenadas

```
N... ORICURVE
N... PO[XH]=(xe, x2, x3, x4, x5)
N... PO[YH]=(ye, y2, y3, y4, y5)
N... PO[ZH]=(ze, z2, z3, z4, z5)
```

Indicação do movimento do segundo ponto de contato da ferramenta e polinômios adicionais das respectivas coordenadas

Parâmetro

`ORICURVE`	`Interpolação da orientação com especificação do movimento de dois pontos de contato da ferramenta.`
`XH YH ZH`	`Identificador das coordenadas do segundo ponto de contato da ferramenta do contorno adicional como curva espacial`
`possíveis polinômios PO[XH]=(xe, x2, x3, x4, x5) PO[YH]=(ye, y2, y3, y4, y5) PO[ZH]=(ze, z2, z3, z4, z5)`	`Além dos respectivos pontos finais, também se pode programar as curvas espaciais com polinômios.`
`xe, ye, ze`	`Pontos finais da curva espacial`
`xi, yi, zi`	`Coeficientes do polinômio de até 5° grau`

Indicação

Identificador XH YH ZH para programação de uma 2ª trajetória de orientação

Os identificadores devem ser selecionados de modo que não resulte em nenhum conflito com outros identificadores dos eixos lineares

Eixos X Y Z

e eixos rotativos como

Ângulo euleriano A2 B2 C2 e ângulo RPY

Vetores de direção A3 B3 C3

Vetores normais de superfície A4 B4 C4 e A5 B5 C5

Vetores de rotação A6 B6 C6 e coordenadas de pontos intermediários A7 B7 C7

ou outros parâmetros de interpolação.

6.3 Polinômios de orientação (PO[ângulo], PO[coordenadas])

Função

Independente de qual interpolação de polinômios do grupo 1 de códigos G está ativo, podem ser programados dois tipos diferentes de polinômios de orientação de até 5º grau em uma transformação de três a cinco eixos.

1. Polinômios para **ângulo:** Ângulo de avanço LEAD, ângulo lateral TILT em relação ao plano que foi fixado pela orientação inicial e orientação final.
2. Polinômios para **coordenadas:** XH, YH, ZH da segunda curva espacial para orientação de ferramenta de um ponto de referência na ferramenta.

Em uma transformação de seis eixos, adicionalmente à orientação de ferramenta, pode ser programada a rotação do vetor de rotação THT com polinômios de até 5º grau para rotações da própria ferramenta.

Sintaxe

Polinômios de orientação de tipo 1 para **ângulo**

```
N… PO[PHI]=(a2, a3, a4, a5)
N… PO[PSI]=(b2, b3, b4, b5)
```

Transformação de três a cinco eixos

Transformação de três a cinco eixos

Polinômios de orientação de tipo 2 para **coordenadas**

```
N… PO[XH]=(xe, x2, x3, x4, x5)
N… PO[YH]=(ye, y2, y3, y4, y5)
N… PO[ZH]=(ze, z2, z3, z4, z5)
```

Identificador para coordenadas da segunda trajetória de orientação para orientação da ferramenta

Nos dois casos também se pode programar um polinômio para a **rotação** em transformações de seis eixos com

```
N… PO[THT]=(c2, c3, c4, c5)
```

Interpolação da rotação relativa à trajetória

ou

```
N… PO[THT]=(d2, d3, d4, d5)
```

interpolação absoluta, relativa e tangencial à alteração da orientação

do vetor de orientação. Isso é possível quando a transformação de um vetor de rotação com um Offset programável e interpolável pelo ângulo de rotação THETA oferecer o suporte necessário.

Significado

`PO[PHI]`	Ângulo no plano entre a orientação inicial e a orientação final
`PO[PSI]`	Ângulo que descreve a inclinação da orientação a partir do plano entre a orientação inicial e orientação final
`PO[THT]`	Ângulo de rotação resultante da rotação do vetor de rotação de um código G do grupo 54 programado com THETA
`PHI`	Ângulo de avanço LEAD
`PSI`	Ângulo lateral TILT
`THETA`	Rotação no sentido da ferramenta em Z
`PO[XH]`	Coordenada X do ponto de referência na ferramenta
`PO[YH]`	Coordenada Y do ponto de referência na ferramenta
`PO[ZH]`	Coordenada Z do ponto de referência na ferramenta

Descrição

Os polinômios de orientação não podem ser programados

- quando as interpolações de Spline ASPLINE, BSPLINE, CSPLINE estiverem ativas.
  Os polinômios do tipo 1 para ângulo de orientação são possíveis para cada tipo de interpolação, exceto para Spline, isto é, possível na interpolação linear com avanço rápido G00 ou com avanço G01
  na interpolação de polinômios com POLY e
  na interpolação de círculos e evolventes com G02, G03, CIP, CT, INVCW e INCCCW
  .
  Em contrapartida, os polinômios do tipo 2 para coordenadas de orientação somente são possíveis se a
  interpolação linear com avanço rápido G00 ou com avanço G01 ou a
  interpolação de polinômios com POLY estiver ativa.
- quando a orientação for interpolada através da interpolação de eixos ORIAXES. Neste caso se pode programar os polinômios diretamente com PO[A] e PO[B] para os eixos de orientação A e B.

Polinômios de orientação do tipo 1 com ORIVECT, ORIPLANE e ORICONxx

Na interpolação de grande circunferência e interpolação de superfície cônica com ORIVECT, ORIPLANE e ORICONxx são possíveis apenas os polinômios de orientação do tipo 1.

Polinômios de orientação do tipo 2 com ORICURVE

Se a interpolação estiver ativa com a curva espacial ORICURVE adicional, os componentes cartesianos do vetor de orientação são interpolados e apenas são possíveis polinômios de orientação do tipo 2.

6.4 Rotações da orientação da ferramenta (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA)

Função

Se em máquinas com ferramenta articulada a orientação da ferramenta também deverá ser possível de ser alterada, então esse bloco também é programado com uma orientação final. Em função da cinemática da máquina também se pode programar o sentido da orientação dos eixos ou a direção de rotação do vetor de orientação THETA. Para estes vetores de rotação podem ser programados vários tipos de interpolação:

- ORIROTA: Ângulo de rotação de um sentido de rotação especificado como absoluto.
- ORIROTR: Ângulo de rotação relativo ao plano entre a orientação inicial e final.
- ORIROTT: Ângulo de rotação relativo à alteração do vetor de orientação.
- ORIROTC: Ângulo de rotação tangencial à tangente da trajetória.

Sintaxe

Somente se o tipo de interpolação `ORIROTA` estiver ativo é que o ângulo de rotação ou o vetor de rotação pode ser programado nos seguintes quatro modos:

1. Diretamente as posições de eixo rotativo `A, B, C`
2. Ângulo euleriano (em graus) através do `A2, B2, C2`
3. Ângulo RPY (em graus) através do `A2, B2, C2`
4. Vetor de direção através do `A3, B3, C3` (ângulo de rotação com `THETA=valor`).

Se `ORIROTR` ou `ORIROTT` estiver ativo, o ângulo de rotação somente pode ser programado diretamente com `THETA`.

Uma rotação também pode ser programada separadamente em um bloco, sem a ocorrência de uma alteração da orientação. Neste caso o `ORIROTR` e o `ORIROTT` não têm nenhum significado. Neste caso o ângulo de rotação sempre é interpretado relativo ao sentido absoluto (`ODIROTA`).

`N... ORIROTA`
`N... ORIROTR`
`N... ORIROTT`
`N... ORIROTC`
	Definir a interpolação do vetor de rotação

`N... A3= B3= C3= THETA=valor`	Definir a rotação do vetor de orientação
`N... PO[THT]=(d2, d3, d4, d5)`	Interpolar o ângulo de rotação com polinômio de 5º grau

Significado

`ORIROTA`	Ângulo de rotação de um sentido de rotação especificado como absoluto
`ORIROTR`	Ângulo de rotação relativo ao plano entre a orientação inicial e a orientação final
`ORIROTT`	Ângulo de rotação como vetor de rotação tangencial à alteração da orientação
`ORIROTC`	Ângulo de rotação como vetor de rotação tangencial à tangente da trajetória
`THETA`	Rotação do vetor de orientação
`THETA=valor`	Ângulo de rotação em graus alcançado no fim do bloco
`THETA=Θe`	Ângulo de rotação com ângulo final Θ_e do vetor de rotação
`THETA=AC(…)`	Comutar para dimensão absoluta por blocos
`THETA=AC(…)`	Comutar para dimensão incremental por blocos
`Θe`	O ângulo final do vetor de rotação está ativo tanto absoluto com G90 como relativo com G91 (dimensão incremental)
`PO[THT]=(...)`	Polinômio para ângulo de rotação

Exemplo de rotações das orientações

Código de programa	**Comentário**
`N10 TRAORI`	`; Ativar transformação de orientação`
`N20 G1 X0 Y0 Z0 F5000`	`; Orientação da ferramenta`
`N30 A3=0 B3=0 C3=1 THETA=0`	`; no sentido Z com ângulo de rotação 0`
`N40 A3=1 B3=0 C3=0 THETA=90`	`; no sentido X e rotação de 90 graus`
`N50 A3=0 B3=1 C3=0 PO[THT]=(180,90)`	`; Orientação`
`N60 A3=0 B3=1 C3=0 THETA=IC(-90)`	`; no sentido Y e rotação de 180 graus`
`N70 ORIROTT`	`; permanece constante rotação de 90 graus`
`N80 A3=1 B3=0 C3=0 THETA=30`	`; Ângulo de rotação relativo à alteração de orientação`
	`; Vetor de rotação em ângulo de 30 graus em relação ao plano X-Y`

Na interpolação do bloco
`N40` o ângulo de rotação é interpolado linearmente do valor inicial de 0 grau até o valor final de 90 graus. No bloco `N50` se altera o ângulo de rotação de 90 graus para 180 graus de acordo com a
parábola $\theta(u) = +90u^2$. No `N60` também pode ser executada uma rotação sem a ocorrência de uma alteração de orientação.

No `N80` se gira a orientação da ferramenta do sentido Y para sentido X. Neste caso a alteração da orientação está no plano X–Y e o vetor de rotação forma um ângulo de 30 graus com este plano.

Descrição

ORIROTA

O ângulo de rotação `THETA` é interpolado relativo a uma direção absoluta definida no espaço. O sentido básico de rotação é realizado através de dados da máquina

ORIROTR

O ângulo de rotação `THETA` é interpretado relativo ao plano fixado pelas orientações inicial e final.

ORIROTT

O ângulo de rotação `THETA` é interpretado relativo à alteração da orientação. Para `THETA=0` o vetor de rotação é interpolado de modo tangencial à alteração da orientação e apenas será diferente do `ORIROTR` se para a orientação foi programado pelo menos um polinômio para o "ângulo de inclinação PSI". Com isso resulta uma alteração da orientação que não percorre no plano. Através de um ângulo de rotação `THETA` programado adicionalmente, o vetor de rotação, por exemplo, pode ser interpolado de modo que ele sempre forma um determinado valor para a alteração da orientação.

ORIROTC

O vetor de rotação é interpolado relativo à tangente da trajetória através de um Offset programável pelo ângulo `THETA`. Para o ângulo de Offset também pode ser programado um polinômio `PO[THT]=(c2, c3, c4, c5)` de até 5º grau.

6.5 Orientações relativas à trajetória

6.5.1 Tipos de orientação relativos à trajetória

Função

Com esta função ampliada não se alcança a orientação relativa apenas no fim do bloco, mas por todo o decurso da trajetória. É realizada a transferência da orientação alcançada no bloco precedente através da interpolação de grande circunferência para a orientação final programada. Basicamente existem duas opções de se programar a orientação desejada de modo relativo à trajetória:

1. A orientação da ferramenta como a rotação da ferramenta é interpolada de modo relativo à trajetória com ORIPATH e ORPATHTS.
2. O vetor de orientação é programado e interpolado da forma usual. Com ORIROTC se posiciona a rotação do vetor de orientação de modo relativo à tangente da trajetória.

Sintaxe

O tipo de interpolação da orientação e da rotação da ferramenta se programa com:

`N... ORIPATH`	Orientação relativa à trajetória
`N... ORIPATHS`	Orientação relativa à trajetória com suavização do decurso de orientação
`N... ORIROTC`	Interpolação do vetor de rotação relativa à trajetória

Uma dobra da orientação ressaltada através de uma esquina no decurso da trajetória pode ser suavizada com `ORIPATHS`. A direção da distância de curso do movimento de retração é programada através do vetor com os componentes `A8=X, B8=Y C8=Z .`

Com `ORIPATH/ORIPATHS` se pode programar diversas referências com a tangente da trajetória através dos três ângulos

- `LEAD=` Indicação do ângulo de avanço relativo à trajetória e superfície
- `TILT=` Indicação do ângulo lateral relativo à trajetória e superfície
- `THETA=` Ângulo de rotação

para o decurso de trajetória total. Ao ângulo de rotação `THETA` também podem ser programados polinômios de até 5º grau com `PO[THT]=(...)`.

Indicação

Fabricante da máquina

Observe as instruções do fabricante da máquina. Através de dados projetáveis de máquina e de ajuste podem ser realizadas outros ajustes para o tipo de orientação relativo à trajetória. Para mais explicações, veja

Literatura:
/FB3/ Manual de funções especiais; Transformação de 3 a 5 eixos (F2),
Capítulo "Orientação"

Significado

A interpolação dos ângulos LEAD e TILT é ajustada de modo diversificado através de dado de máquina.

- A referência da orientação da ferramenta programada com LEAD e TILT é conservada por todo o bloco.
- Ângulo de avanço LEAD: Rotação em torno da direção perpendicular à tange e vetor normal TILT: Rotação da orientação em torno do vetor normal.
- Ângulo de avanço LEAD: Rotação em torno da direção perpendicular à tange e vetor normal do ângulo lateral TILT: Rotação da orientação em torno da direção da tangente da trajetória.
- Ângulo de rotação THETA: Rotação da ferramenta em torno de si mesma com um terceiro eixo rotativo extra como eixo de orientação na transformação de seis eixos.

Indicação

A orientação relativa à trajetória não é permitida junto com OSC, OSS, OSSE, OSD, OST

A interpolação de orientação relativa à trajetória ORIPATH ou ORIPATHS e ORIOTC não pode ser programada junto com a suavização do decurso de orientação com um dos códigos G do grupo 34. Para isso o OSOF deve estar ativo.

6.5.2 Rotação da orientação da ferramenta relativa à trajetória (ORIPATH, ORIPATHS, ângulo de rotação)

Função

Em uma transformação de seis eixos, a ferramenta também pode ser girada em torno de si mesma com um terceiro eixo rotativo para qualquer orientação de ferramenta no espaço. Na rotação da orientação de ferramenta relativa à trajetória com ORIPATH ou ORIPATHS, se pode programar a rotação adicional através do ângulo de rotação THETA. De modo alternativo, se pode programar os ângulos LEAD e TILT através de um vetor que está perpendicular à direção da ferramenta no plano.

Fabricante da máquina

Observe as instruções do fabricante da máquina. Através de dado de máquina se pode ajusta de forma diferente a interpolação dos ângulos LEAD e TILT.

Sintaxe

Rotação da orientação da ferramenta e da própria ferramenta

O tipo de orientação de ferramenta relativo à trajetória é ativado com ORIPATH ou ORIPATHS.

`N... ORIPATH`	Ativar o tipo de orientação relativo à trajetória
`N... ORIPATHS`	Ativação do tipo de orientação relativo à trajetória com suavização do decurso de orientação

Ativação dos três possíveis ângulos com efeito de rotação:

`N... LEAD=`	Ângulo para orientação programada relativa ao vetor normal de superfície
`N... TILT=`	Ângulo para orientação programada no plano perpendicular à tangente da trajetória relativa ao vetor normal de superfície
`N... THETA=`	Ângulo de rotação relativo à alteração da orientação em torno da direção de ferramenta do terceiro eixo rotativo

Os valores do ângulo no fim do bloco são programados com `LEAD=valor`, `TILT=valor` ou `THETA=valor`. Adicionalmente aos ângulos constantes podem ser programados polinômios de até 5º grau para todos os três ângulos.

`N... PO[PHI]=(a2, a3, a4, a5)`	Polinômio para ângulo de avanço LEAD
`N... PO[PSI]=(b2, b3, b4, b5)`	Polinômio para ângulo lateral TILT
`N... PO[THT]=(d2, d3, d4, d5)`	Polinômio para ângulo de rotação THETA

Na programação podem ser descartados os coeficientes de polinômio mais altos, que são igual a zero. O exemplo `PO[PHI]=a2` resulta em uma parábola para o ângulo de avanço `LEAD`.

Significado

Orientação de ferramenta relativa à trajetória

`ORIPATH`	Orientação da ferramenta relativa à trajetória
`ORIPATHS`	A orientação de ferramenta relativa à dobra de trajetória no decurso de orientação é suavizada
`LEAD`	Ângulo relativo ao vetor normal de superfície, no plano fixado pela tangente da trajetória e vetor normal de superfície
`TILT`	Rotação da orientação em torno do sentido Z ou rotação em torno da tangente da trajetória
`THETA`	Rotação em torno do sentido da ferramenta para Z
`PO[PHI]`	Polinômio de orientação para ângulo de avanço LEAD
`PO[PSI]`	Polinômio de orientação para ângulo lateral TILT
`PO[THT]`	Polinômio de orientação para ângulo de rotação THETA

Indicação

Ângulo de rotação THETA

Para a rotação da ferramenta com terceiro eixo rotativo como eixo de orientação em torno de si mesma é necessária uma transformação de seis eixos.

6.5.3 Interpolação relativa à trajetória da rotação da ferramenta (ORIROTC, THETA)

Função

Interpolação com vetores de rotação

À rotação de ferramenta relativa à tangente de trajetória programada com ORIROTC se pode interpolar o vetor de direção com um Offset programável através do ângulo de rotação THETA. Neste caso se pode programar um polinômio de até 5º grau com PO[THT] para o ângulo de Offset.

Sintaxe

`N... ORIROTC`	Posicionar a rotação da ferramenta relativa à tangente de trajetória
`N... A3= B3= C3= THETA=valor`	Definir a rotação do vetor de orientação
`N... A3= B3= C3= PO[THT]=(c2, c3, c4, c5)`	Interpolar o ângulo Offset com polinômio de até 5º grau

Uma rotação também pode ser programada separadamente em um bloco, sem a ocorrência de uma alteração da orientação.

Significado

Interpolação da rotação da ferramenta relativa à trajetória na transformação de seis eixos

`ORIROTC`	posicionar vetor de rotação tangencial à tangente de trajetória
`THETA=valor`	Ângulo de rotação em graus alcançado no fim do bloco
`THETA=θe`	Ângulo de rotação com ângulo final Θ_e do vetor de rotação
`THETA=AC(…)`	Comutar para dimensão absoluta por blocos
`THETA=IC(…)`	Comutar para dimensão incremental por blocos
`PO[THT]=(c2, c3, c4, c5)`	Interpolar o ângulo de Offset com polinômio de 5º grau

Indicação

Interpolação do vetor de rotação ORIROTC

Se a rotação da ferramenta relativa à tangente de trajetória também deve ser posicionada contra o sentido da orientação da ferramenta, então isso será possível apenas com uma transformação de seis eixos.

Com ORIROTC ativo

O vetor de rotação ORIROTA não pode ser programado. No caso de uma programação, será emitido o ALARME 14128 "Programação absoluta da rotação da ferramenta com ORIROTC ativo".

Sentido de orientação da ferramenta na transformação de três a cinco eixos

O sentido de orientação da ferramenta pode ser programado da forma usual, através de ângulo euleriano e ângulo RPY ou através de vetores de direção, como na transformação de três a cinco eixos. Também são possíveis as alterações de orientação da ferramenta no espaço através da programação da interpolação de grande circunferência ORIVECT, interpolação linear dos eixos de orientação ORIAXES, todas interpolações sobre uma superfície periférica cônica ORICONxx, e também a interpolação adicional à curva espacial com dois pontos de contato da ferramenta ORICURVE.

`G....`	Indicação do tipo de movimento dos eixos rotativos
`X Y Z`	Indicação dos eixos lineares
`ORIAXES`	Interpolação linear dos eixos da máquina ou dos eixos de orientação
`ORIVECT`	Interpolação de grande circunferência (idêntico ao ORIPLANE)
`ORIMKS`	Rotação no sistema de coordenadas da máquina
`ORIWKS`	Rotação no sistema de coordenadas da peça de trabalho
	Veja a descrição no cap. Rotações da orientação da ferramenta
`A= B= C=`	Programação da posição de eixo da máquina
`ORIEULER`	Programação da orientação através de ângulo euleriano
`ORIRPY`	Programação da orientação através de ângulo RPY
`A2= B2= C2=`	Programação de ângulo dos eixos virtuais
`ORIVIRT1`	Programação da orientação através de eixos virtuais de orientação
`ORIVIRT2`	(definição 1), definição conforme MD $MC_ORIAX_TURN_TAB_1
	(definição 2), definição conforme MD $MC_ORIAX_TURN_TAB_2
`A3= B3= C3=`	Programação do vetor de direção do eixo de sentido
`ORIPLANE`	Interpolação no plano (interpolação de grande circunferência)
`ORICONCW`	Interpolação sobre uma superfície periférica cônica no sentido horário
`ORICONCCW`	Interpolação sobre uma superfície periférica cônica no sentido anti-horário
`ORICONTO`	Interpolação sobre uma superfície periférica cônica de transição tangencial
`A6= B6= C6=`	Programação do eixo de rotação do cone (vetor normalizado)
`NUT=ângulo`	Ângulo de abertura do cone em graus
`NUT=+179`	Ângulo de deslocamento menor ou igual a 180 graus
`NUT=-181`	Ângulo de deslocamento maior ou igual a 180 graus
`ORICONIO`	Interpolação sobre uma superfície periférica cônica
`A7= B7= C7=`	Orientação intermediária (programação como vetor normalizado)
`ORICURVE XH YH ZH p. ex. com polinômios PO[XH]=(xe, x2, x3, x4, x5)`	Interpolação da orientação com especificação do movimento de dois pontos de contato da ferramenta. Além dos respectivos pontos finais também se pode programar polinômios às curvas espaciais extras.

Indicação

Se a orientação de ferramenta com ORIAXES ativo for interpolada através dos eixos de orientação, então o posicionamento relativo à trajetória do ângulo de rotação somente será realizado no fim do bloco.

6.5.4 Suavização do decurso de orientação (ORIPATHS A8=, B8=, C8=)

Função

Nas alterações de orientação com aceleração constante no contorno existem interrupções nos movimentos de percurso que ocorrem principalmente nos cantos (esquinas) de um contorno, que não são desejadas. A dobra resultante disso no decurso da orientação pode ser suavizada através da inserção de um bloco intermediário próprio. A alteração de orientação é realizada com aceleração constante se durante a reorientação o ORIPATHS também estiver ativo. Nesta fase pode ser realizado um movimento de retração da ferramenta.

Fabricante da máquina

Observe as instruções do fabricante da máquina sobre eventuais dados de máquina e dados de ajuste pré-definidos com os quais se ativa esta função.

Através de dado de máquina se pode ajustar como o vetor de retração deve ser interpretado:

1. No sistema de coordenadas da ferramenta as coordenadas em Z estão definidas através do sentido da ferramenta.
2. No sistema de coordenadas da peça as coordenadas em Z estão definidas através do plano ativo.

Para mais explanações sobre a função "Orientação relativa à trajetória", veja
Literatura: /FB3/ Manual de funções especiais; Transformações de 3 a 5 eixos (F2)

Sintaxe

Para orientações de ferramenta relativas à trajetória inteira, realizadas constantemente, são necessárias outras informações de programação em um canto do contorno. O sentido e a distância de curso deste movimento são programados através do vetor com os componentes A8=X, B8=Y, C8=Z:

```
N... ORIPATHS A8=X B8=Y C8=Z
```

Significado

`ORIPATHS`	Orientação de ferramenta relativa à trajetória, uma dobra é suavizada no decurso da orientação.
`A8= B8= C8=`	Componentes de vetor para sentido e distância de curso
`X, Y, Z`	Movimento de retração no sentido da ferramenta

Indicação

Programação do vetor de direção A8, B8, C8

Se a distância deste vetor for igual a zero, não se realiza nenhum movimento de retração.

ORIPATHS

A orientação de ferramenta relativa à trajetória é ativada com ORIPATHS. Caso contrário se transporta a orientação através de interpolação linear de grande circunferência da orientação inicial para a orientação final.

6.6 Compressão da orientação (COMPON, COMPCURV, COMPCAD)

Função

Os programas NC que possuem uma transformação de orientação (TRAORI) ativa, cuja orientação foi programada através de vetores de direção, podem ser comprimidos com a conservação das tolerâncias especificadas.

Indicação

O movimento de orientação somente é comprimido com a interpolação de grande circunferência ativa e, dessa forma, ele é dependente de códigos G para a interpolação de orientação. Este também pode ser ajustado como a distância de curso máxima e uma tolerância admissível para cada eixo ou para o avanço de trajetória para a função de compressor através de dados da máquina. Observe as instruções do fabricante da máquina.

Programação

Orientação da ferramenta

Se uma transformação de orientação (TRAORI) estiver ativa, a orientação de ferramenta em máquinas de 5 eixos pode ser programada da seguinte forma (independente da cinemática):

- Programação do **vetor** de direção através do:

  ```
  A3=<...> B3=<...> C3=<...>
  ```

- Programação do **ângulo euleriano** ou **ângulo RPY** através de:

  ```
  A2=<...> B2=<...> C2=<...>
  ```

Rotação da ferramenta

Para máquinas de **6 eixos** também se pode programar a rotação da ferramenta para orientação de ferramenta.

A programação do ângulo de rotação é realizada com:

```
THETA=<...>
```

Veja " Rotações da orientação da ferramenta (Página 350) ".

Indicação

Os blocos NC em que foi programada uma rotação adicional, somente podem ser comprimidos se o ângulo de rotação se alterar **linearmente**. Isto significa que para o ângulo de rotação não pode ser programado nenhum polinômio com PO[THT]=(...).

Forma geral de um bloco NC compressível

Por isso que a forma geral de um bloco NC compressível pode se apresentar da seguinte maneira:

```
N... X=<...> Y=<...> Z=<...> A3=<...> B3=<...> C3=<...> THETA=<...> F=<...>
```

ou

```
N... X=<...> Y=<...> Z=<...> A2=<...> B2=<...> C2=<...> THETA=<...> F=<...>
```

Indicação

Os valores de posição podem ser especificados de modo direto (p. ex. X90) ou indireto através da atribuição de parâmetros (p. ex. X=R1*(R2+R3)).

Programação da orientação de ferramenta através de posições de eixo rotativo

A orientação da ferramenta também pode ser especificada através de posições de eixo rotativo, por exemplo, na seguinte forma:

```
N... X=<...> Y=<...> Z=<...> A=<...> B=<...> C=<...> THETA=<...> F=<...>
```

Neste caso a compressão é executada de dois modos diferentes, em função de ser executada uma interpolação de grande circunferência ou não. Se não ocorre nenhuma interpolação de grande circunferência, então a alteração de orientação comprimida é representada através de polinômios axiais para eixos rotativos, como no procedimento usual.

Precisão de contorno

Em função do modo de compressão ajustado (MD20482 $MC_COMPRESSOR_MODE), durante a compressão, para eixos geométricos e eixos de orientação são ativadas as tolerâncias específicas de eixo projetadas (MD33100 $MA_COMPRESS_POS_TOL) ou as seguintes tolerâncias específicas de canal ajustadas através de dados de ajuste:

SD42475 $SC_COMPRESS_CONTUR_TOL (desvio máximo do contorno)

SD42476 $SC_COMPRESS_ORI_TOL (desvio angular máximo para orientação da ferramenta)

SD42477 $SC_COMPRESS_ORI_ROT_TOL (desvio máximo do ângulo de rotação da ferramenta) (disponível apenas em máquinas de 6 eixos).

Literatura:
Manual de funções básicas; Transformação de 3 a 5 eixos (F2),
Capítulo: "Compressão da orientação"

Ativação / desativação

As funções de compressor são ativadas através dos códigos G COMPON, COMPCURV e COMPCAD.

A função de compressor é encerrada com COMPOF.

Veja " Compressão de blocos NC (COMPON, COMPCURV, COMPCAD) (Página 251) ".

Indicação

O movimento de orientação somente é comprimido com a interpolação de grande circunferência ativa (isto é, a alteração da orientação da ferramenta é realizada no plano fixado pelas orientações inicial e final).

Uma interpolação de grande circunferência é executada sob as seguintes condições:

- MD21104 $MC_ORI_IPO_WITH_G_CODE = 0,
  o ORIWKS está ativo e
  a orientação é programada através de vetores (com A3, B3, C3 ou A2, B2, C2).
- MD21104 $MC_ORI_IPO_WITH_G_CODE = 1 e o
  ORIVECT ou ORIPLANE está ativo.

  A orientação da ferramenta pode ser programada como vetor de direção ou com posições de eixo rotativo. Se um dos códigos G ORICONxx ou ORICURVE estiver ativo ou se foram programados polinômios para o ângulo de orientação (PO[PHI] e PO[PSI]), não será executada nenhuma interpolação de grande circunferência.

Exemplo

No exemplo de programação a seguir se comprime uma circunferência que é aproximada por um polinômio. A orientação da ferramenta se movimenta de modo sincronizado sobre uma superfície periférica cônica. Mesmo que as alterações de orientação estejam programadas em seqüência, mas não são contínuas, a função do compressor gera um movimento suave da orientação.

```
Programação                                  Comentário
DEF INT QUANTIDADE=60
DEF REAL RAIO=20
DEF INT COUNTER
DEF REAL ANGULO
N10 G1 X0 Y0 F5000 G64

$SC_COMPRESS_CONTUR_TOL=0.05                 ; Desvio máximo do contorno = 0.05 mm
$SC_COMPRESS_ORI_TOL=5                       ; Desvio máximo da orientação = 5° grau

TRAORI
COMPCURV
                                             ; Se realiza um movimento em círculo que é
                                               formado por polinômios. A orientação
                                               movimenta-se sobre um cone em torno do eixo
                                               Z com um ângulo de abertura de 45 graus.
N100 X0 Y0 A3=0 B3=-1 C3=1
N110 FOR COUNTER=0 TO QUANTIDADE
N120 ANGULO=360*COUNTER/QUANTIDADE
N130 X=RAIO*cos(ANGULO) Y=RAIO*sin(ANGULO)
A3=sin(ANGULO) B3=-cos(ANGULO) C3=1
N140 ENDFOR
```

6.7 Suavização do decurso de orientação (ORISON, ORISOF)

Função

Com a função "Suavização do decurso de orientação (ORISON)" é possível suavizar as variações da orientação por vários blocos. Dessa forma é obtido um percurso suave tanto da orientação como do contorno.

Pré-requisito

A função "Suavização do decurso de orientação (ORISON)" somente está disponíveis em sistemas com transformação de 5/6 eixos.

Sintaxe

```
ORISON
...
ORISOF
```

Significado

`ORISON`:	Suavização do decurso de orientação ON	
	Efeito:	modal
`ORISOF`:	Suavização do decurso de orientação OFF	
	Efeito:	modal

Dados de ajuste

A suavização do decurso de orientação é realizado sob o cuidado de manter:

- uma tolerância máxima pré-definida (desvio angular máximo da orientação da ferramenta dado em graus)

  e

- um percurso máximo pré-definido.

Estas definições são definidas através de dados de ajuste:

- SD42678 $SC_ORISON_TOL (tolerância para suavização do decurso de orientação)
- SD42680 O$SC_ORISON_DIST (percurso para a suavização do decurso de orientação)

Exemplo

```
Código de programa                Comentário
...
TRAORI()                          ; Ativação da transformação de orientação.
ORISON                            ; Ativação da suavização de orientação.
$SC_ORISON_TOL=1.0                ; Tolerância da suavização de orientação
                                    = 1,0 grau.
G91
X10 A3=1 B3=0 C3=1
X10 A3=-1 B3=0 C3=1
X10 A3=1 B3=0 C3=1
X10 A3=-1 B3=0 C3=1
X10 A3=1 B3=0 C3=1
X10 A3=-1 B3=0 C3=1
X10 A3=1 B3=0 C3=1
X10 A3=-1 B3=0 C3=1
X10 A3=1 B3=0 C3=1
X10 A3=-1 B3=0 C3=1
...
ORISOF                            ; Desativação da suavização de orientação.
...
```

A orientação é girada em 90 graus no plano XZ de -45 até +45 graus. Através da suavização do decurso de orientação, esta não alcança mais os valores angulares máximos de -45 e +45 graus, respectivamente.

Outras informações

Número de blocos

A suavização do decurso de orientação é realizada através de um número de blocos configurado, que consta no dado de máquina MD28590 $MC_MM_ORISON_BLOCKS.

Indicação

Quando a suavização do decurso de orientação é ativada com `ORISON`, sem haver memória de bloco suficiente configurada para este fim (MD28590 < 4), então aparece uma mensagem de alarme e a função não pode ser executada.

Distâncias de percurso de bloco máximas

O decurso de orientação somente é suavizado nos blocos cujo percurso for menor que a distância de percurso de bloco máxima (MD20178 $MC_ORISON_BLOCK_PATH_LIMIT). Os blocos com percursos mais extensos interrompem a suavização e são executadas como foram programadas.

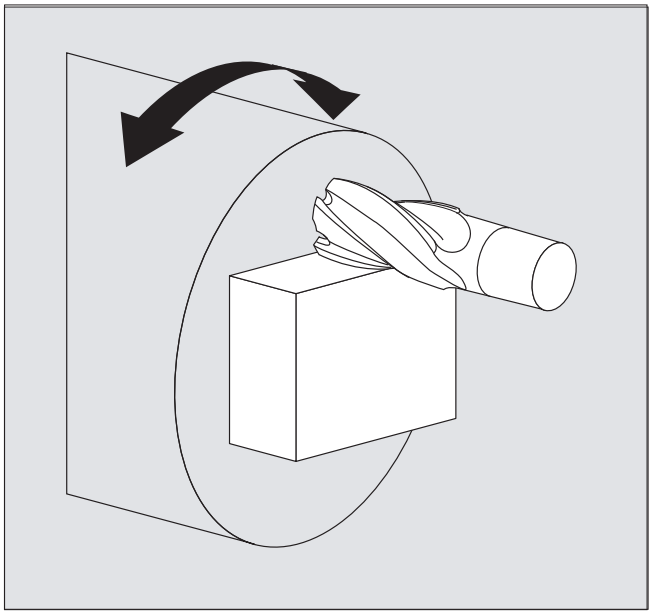
6.8 Transformação cinemática

6.8.1 Fresamento em peças torneadas (TRANSMIT)

Função

A função TRANSMIT possibilita as seguintes tarefas:

- Usinagem de topo em peças torneadas na fixação de torneamento (furos, contornos).
- Para programação destas operações de usinagem se pode usar um sistema de coordenadas cartesiano.
- O comando transforma os movimentos de deslocamento programados do sistema de coordenadas cartesiano em movimentos de deslocamento dos eixos reais da máquina (caso padrão):
  - Eixo rotativo
  - Eixo de penetração vertical ao eixo de rotação
  - Eixo longitudinal paralelo ao eixo de rotação
  - Os eixos lineares estão perpendiculares entre si.
- O deslocamento do centro da ferramenta relativo ao centro de rotação é permitido.
- O controle da velocidade considera os limites definidos para os movimentos de rotação.



Tipos de transformação TRANSMIT

Para operações TRANSMIT existem duas formas ajustáveis:

- `TRANSMIT` em caso padrão com (TRAFO_TYPE_n = 256)
- `TRANSMIT` com eixo linear Y adicional (TRAFO_TYPE_n = 257)

Por exemplo, o tipo de transformação 257 ampliado pode ser utilizado para compensar correções de fixação de uma peça com eixo Y real.

Sintaxe

`TRANSMIT` ou `TRANSMIT(n)`

`TRAFOOF`

Eixo rotativo

O eixo rotativo não pode ser programado, pois ele é ocupado por um eixo geométrico e dessa forma não poderá ser programado diretamente como eixo de canal.

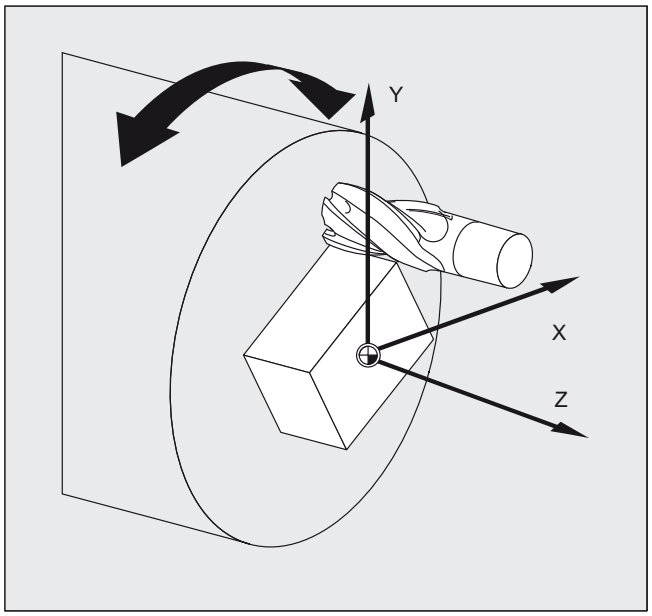
Significado

`TRANSMIT`:	Ativa a primeira função TRANSMIT acordada. Esta função também é denominada de transformação polar.
`TRANSMIT(n)`:	Ativa a n função TRANSMIT acordada; o n pode ser no máximo 2 (TRANSMIT(1) corresponde ao TRANSMIT).
`TRAFOOF`:	Desativa uma transformação ativa
`OFFN`:	Offset normal de contorno: Distância da usinagem frontal do contorno de referência

Indicação

Uma transformação ativa `TRANSMIT` também é desativada se uma das demais transformações estiver ativa no respectivo canal (p. ex. `TRACYL`, `TRAANG`, `TRAORI`).

Exemplo



Código de programa	**Comentário**
N10 T1 D1 G54 G17 G90 F5000 G94	; Seleção da ferramenta
N20 G0 X20 Z10 SPOS=45	; Aproximação da posição de saída
N30 TRANSMIT	; Ativação da função TRANSMIT
N40 ROT RPL=-45	; Ajustar Frame
N50 ATRANS X-2 Y10	
N60 G1 X10 Y-10 G41 OFFN=1OFFN	; Desbaste do quadrado; sobremetal de 1 mm
N70 X-10	
N80 Y10	
N90 X10	
N100 Y-10	
N110 G0 Z20 G40 OFFN=0	; Troca de ferramentas
N120 T2 D1 X15 Y-15	
N130 Z10 G41	
N140 G1 X10 Y-10	; Acabamento do quadrado
N150 X-10	
N160 Y10	
N170 X10	
N180 Y-10	
N190 Z20 G40	; Desativação de Frame
N200 TRANS	
N210 TRAFOOF	
N220 G0 X20 Z10 SPOS=45	; Aproximação da posição de saída
N230 M30	

Descrição

Pólo

Para passar pelo pólo existem duas possibilidades:

- Deslocamento isolado do eixo linear
- Deslocamento no pólo com rotação do eixo rotativo no pólo e saída do pólo

A seleção é realizada através do MD 24911 e 24951.

TRANSMIT com eixo linear Y adicional (tipo de transformação 257):

Esta variante de transformação polar aproveita a redundância em uma máquina com um eixo linear extra para executar uma correção de ferramenta mais aprimorada. Então para o segundo eixo linear se aplica:

- uma área de trabalho menor e
- que o segundo eixo linear não seja utilizado na execução do programa de peça.

Para o programa de peça e a atribuição dos respectivos eixos no BCS ou MCS são requeridos determinados ajustes de dados de máquina, veja

Literatura

/FB2/ Manual de funções ampliadas; Transformações cinemáticas (M1)

6.8.2 Transformação de superfície cilíndrica (TRACYL)

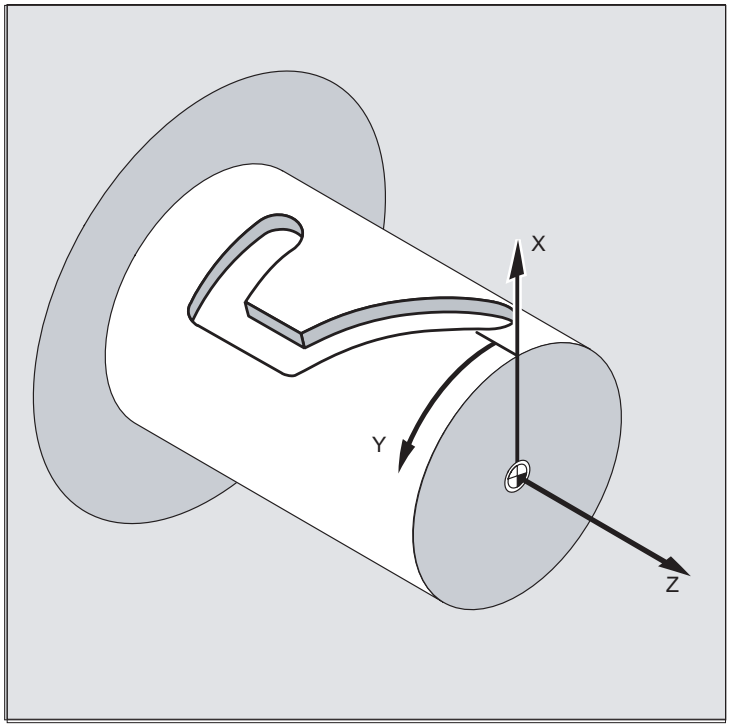
Função

A transformação de curvas da superfície periférica do cilindro TRACYL possibilita as seguintes tarefas:

Usinagem de

- ranhuras longitudinais em corpos cilíndricos,
- ranhuras transversais em corpos cilíndricos,
- ranhuras de qualquer direção em corpos cilíndricos.

A trajetória das ranhuras é programada relativa à superfície desenvolvida do cilindro.



Tipos de transformação TRACYL

A transformação das coordenadas da superfície periférica do cilindro existe em três formas:

- `TRACYL` sem correção da parede da ranhura: (TRAFO_TYPE_n=512)
- `TRACYL` com correção da parede da ranhura: (TRAFO_TYPE_n=513)
- `TRACYL` com eixo linear adicional e com correção da parede da ranhura: (TRAFO_TYPE_n=514)
  A correção da parede da ranhura é parametrizada com `TRACYL` através do terceiro parâmetro.

Para a transformação de curvas da superfície periférica com correção da parede da ranhura o eixo utilizado para a correção deveria estar com valor zero (y=0), para que a ranhura seja produzida de forma centralizada na linha de centro da ranhura.

Utilização de eixo

Os seguintes eixos não podem ser utilizados como eixo de posicionamento ou eixo oscilante:

- o eixo geométrico no sentido da superfície periférica do cilindro (eixo Y)
- o eixo linear adicional na correção da parede da ranhura (eixo Z)

Sintaxe

`TRACYL(d)` ou `TRACYL(d, n)` ou

para tipo de transformação 514

```
TRACYL(d, n, correção da parede da ranhura)
TRAFOOF
```

Eixo rotativo

O eixo rotativo não pode ser programado, pois ele é ocupado por um eixo geométrico e dessa forma não poderá ser programado diretamente como eixo de canal.

Significado

`TRACYL(d)`	Ativa a primeira função TRACYL acordada nos dados de máquina de canal. Parâmetro d para diâmetro de trabalho.
`TRACYL (d, n)`	Ativa a função TRACYL n. acordada nos dados de máquina de canal. O n pode ser no máximo 2, o TRACYL(d,1) corresponde ao TRACYL(d).
`D`	Valor do diâmetro de trabalho. O diâmetro de trabalho é a dupla distância entre a ponta da ferramenta e o centro de rotação. Este diâmetro sempre deve ser especificado e deve ser maior que 1.
`n`	2º parâmetro opcional para bloco de dados TRACYL 1 (preselecionado) ou 2.
`Correção da parede da ranhura`	3º parâmetro opcional cujo valor para TRACYL é preselecionado dos dados da máquina. Faixa de valores: 0: Tipo de transformação 514 sem correção da parede da ranhura usual 1: Tipo de transformação 514 com correção da parede da ranhura
`TRAFOOF`	Transformação desativada (BCS e MCS são idênticos novamente).
`OFFN`	Offset normal de contorno: Distância da parede da ranhura até o contorno de referência programado

Indicação

Uma transformação `TRACYL` ativa também é desativada se o respectivo canal uma das demais transformações estiver ativa (p. ex. `TRANSMIT`, `TRAANG`, `TRAORI`).

Exemplo: Definição da ferramenta

O exemplo a seguir é apropriado para testar a parametrização da transformação de cilindro TRACYL:

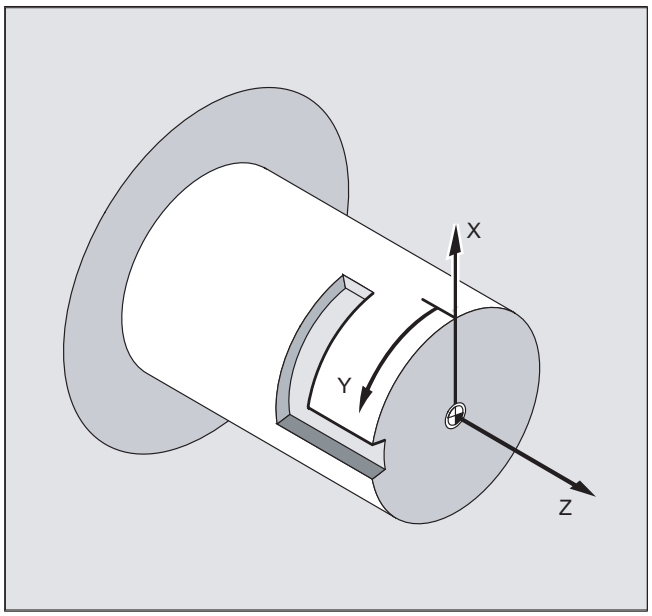
Código de programa	**Comentário**	
Parâmetro da ferramenta	Significado	Observação
Número (DP)		
$TC_DP1[1,1]=120	Tipo de ferramenta	Fresa
$TC_DP2[1,1]=0	Posição de corte	apenas para ferramentas de tornear

Código de programa	**Comentário**	
Geometria	Correção de comprimentos	
$TC_DP3[1,1]=8.	Vetor de correção de comprimento	Processamento por tipo
$TC_DP4[1,1]=9.		e plano
$TC_DP5[1,1]=7.		

Código de programa	**Comentário**	
Geometria	Raio	
$TC_DP6[1,1]=6.	Raio	Raio da ferramenta
$TC_DP7[1,1]=0	Largura de ranhura b para serra de ranhuras, raio de arredondamento para fresas	
$TC_DP8[1,1]=0	Ressalto k	apenas para serra para ranhuras
$TC_DP9[1,1]=0		
$TC_DP10[1,1]=0		
$TC_DP11[1,1]=0	Ângulo para fresas cônicas	

Código de programa	**Comentário**	
Desgaste	Correção de comprimento e raio	
$TC_DP12[1,1]=0	O parâmetros restantes até $TC_DP24=0	Dimensão básica/Adaptador

Exemplo: Produção de uma ranhura em forma de gancho



Ativação da transformação de superfície cilíndrica:

Código de programa	Comentário
N10 T1 D1 G54 G90 F5000 G94	; Seleção de ferramenta, compensação de fixação
N20 SPOS=0	; Aproximação da posição de saída
N30 G0 X25 Y0 Z105 CC=200	
N40 TRACYL (40)	; Ativar a transformação de curvas da superfície periférica do cilindro
N50 G19	; Seleção de plano

Produção de ranhura em forma de gancho:

Código de programa	Comentário
N60 G1 X20	; Penetrar a ferramenta até a base da ranhura
N70 OFFN=12	; Definir distância de paredes da ranhura de 12 mm relativa à linha de centro da ranhura
N80 G1 Z100 G42	; Aproximação da parede direita da ranhura
N90 G1 Z50	; Segmento de ranhura paralelo ao eixo do cilindro
N100 G1 Y10	; Segmento de ranhura paralelo à superfície periférica
N110 OFFN=4 G42	; Aproximação da parede esquerda da ranhura; definição da distância de paredes da ranhura de 4 mm relativa à linha de centro da ranhura
N120 G1 Y70	; Segmento de ranhura paralelo à superfície periférica

Código de programa	Comentário
N130 G1 Z100	; Segmento de ranhura paralelo ao eixo do cilindro
N140 G1 Z105 G40	; Afastamento da parede da ranhura
N150 G1 X25	; Afastamento
N160 TRAFOOF	
N170 G0 X25 Y0 Z105 CC=200	; Aproximação da posição de saída
N180 M30	

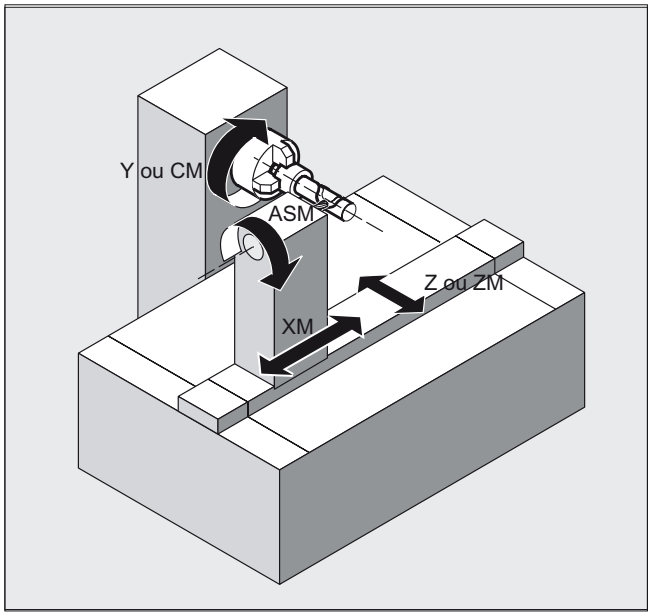
Descrição

Sem correção da parede da ranhura (tipo de transformação 512):

O comando transforma os movimentos de percurso programados do sistema de coordenadas do cilindro em movimentos de deslocamento dos eixos reais da máquina:

- Eixo rotativo
- Eixo de penetração vertical ao eixo de rotação
- Eixo longitudinal paralelo ao eixo de rotação

Os eixos lineares estão perpendiculares entre si. O eixo de penetração atravessa o eixo rotativo.

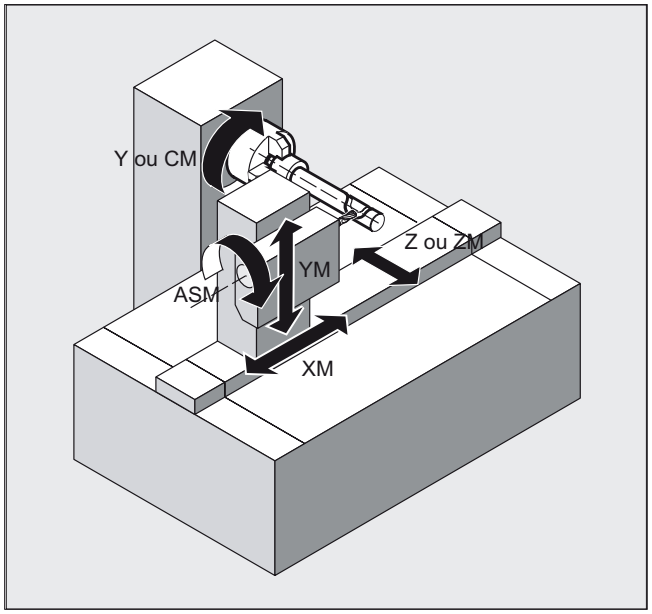


Com correção da parede da ranhura (tipo de transformação 513):

Cinemática como mencionado acima, mas adicionada do eixo longitudinal paralelo ao sentido periférico

Os eixos lineares estão perpendiculares entre si.

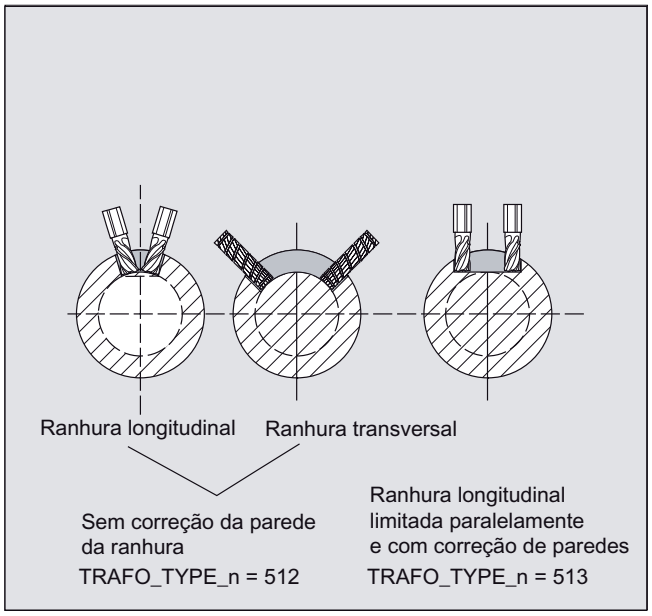
O controle da velocidade considera os limites definidos para os movimentos de rotação.



Secção transversal da ranhura

Na configuração de eixo 1 as ranhuras longitudinais relativas ao eixo rotativo somente estão limitadas paralelamente se a largura da ranhura corresponder exatamente ao raio da ferramenta.

As ranhuras paralelas à área periférica (ranhuras transversais) não são paralelas no início e no fim.


Ranhura longitudinal
Ranhura transversal
Sem correção da parede da ranhura
TRAFO_TYPE_n = 512
Ranhura longitudinal limitada paralelamente e com correção de paredes
TRAFO_TYPE_n = 513

Com eixo linear adicional e com correção da parede da ranhura (tipo de transformação 514):

Esta variante de transformação aproveita a redundância em uma máquina com um eixo linear extra para executar uma correção de ferramenta mais aprimorada. Então para o segundo eixo linear se aplica:

- uma área de trabalho menor e
- que o segundo eixo linear não seja utilizado na execução do programa de peça.

Para o programa de peça e a atribuição dos respectivos eixos no BCS ou MCS são requeridos determinados ajustes de dados de máquina, veja

Literatura

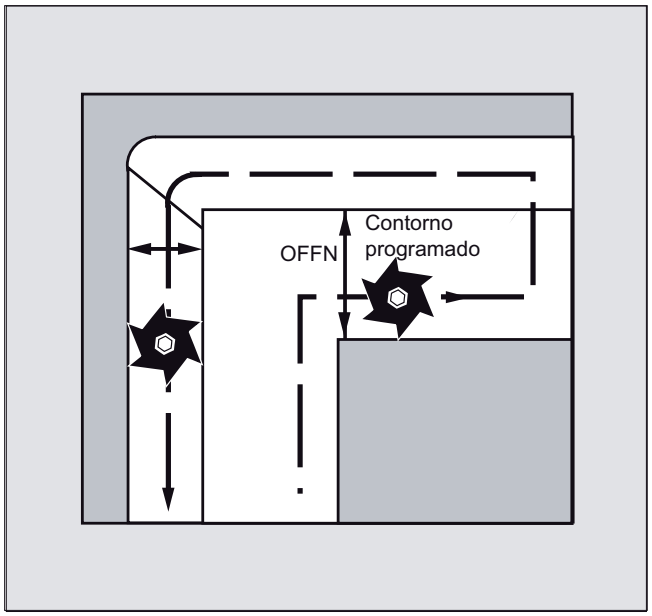
/FB2/ Manual de funções ampliadas; Transformações cinemáticas (M1)

Offset normal de contorno OFFN (tipo de transformação 513)

Para fresar ranhuras com TRACYL, se programa no

- programa de peças a linha de centro da ranhura,
- a **metade da ranhura** através do OFFN.

O OFFN somente é ativado com a seleção do correção do raio da ferramenta, para evitar danos na parede da ranhura. **Além disso o OFFN deveria ser >=raio da ferramenta para evitar danos à parede da ranhura do lado oposto.**



Um programa de peça para fresar uma ranhura normalmente é constituído pelos seguintes passos:

1. Selecionar ferramenta
2. Selecionar TRACYL
3. Selecionar o deslocamento de coordenadas (FRAME) adequado
4. Posicionar
5. Programar OFFN
6. Selecionar a correção do raio da ferramenta
7. Bloco de aproximação (entrada da correção do raio da ferramenta e aproximação da parede da ranhura)
8. Contorno da linha de centro da ranhura
9. Desselecionar a correção do raio da ferramenta
10. Bloco de afastamento (saída da correção do raio da ferramenta e afastamento da parede da ranhura)
11. Posicionar
12. TRAFOOF
13. Selecionar novamente o deslocamento de coordenadas (FRAME) original

Particularidades

- Seleção da correção do raio da ferramenta:

  A correção do raio da ferramenta não é programada em função da parede da ranhura, mas relativa à linha de centro da ranhura. Para que a ferramenta percorra à esquerda da parede da ranhura especifica-se G42 (ao invés de G41). Evitamos isso se a largura da ranhura for especificada com um sinal negativo no OFFN.

- O OFFN com TRACYL atua de modo diferente como sem TRACYL. Visto que OFFN também é incluso sem TRACYL com a correção do raio da ferramenta ativa, então o OFFN deve ser zerado novamente após o TRAFOOF.

- É possível alterar o OFFN durante o programa de peça. Com isso a linha de centro da ranhura poderia ser deslocada do centro (veja a figura).

- Ranhuras de guia:

  Com TRACYL não é produzida a mesma ranhura de guia como se fosse gerada com uma ferramenta cujo diâmetro eqüivale à largura da ranhura. A princípio não é possível produzir a mesma geometria de parede de ranhura com uma ferramenta cilíndrica pequena do que com uma maior. O TRACYL minimiza o erro. Para evitar problemas de precisão, o raio da ferramenta de ser apenas um pouco menor que a metade da largura da ranhura.

Indicação

OFFN e correção do raio da ferramenta

Com TRAFO_TYPE_n = 512 o valor no OFFN atua como sobremetal para a correção do raio da ferramenta.

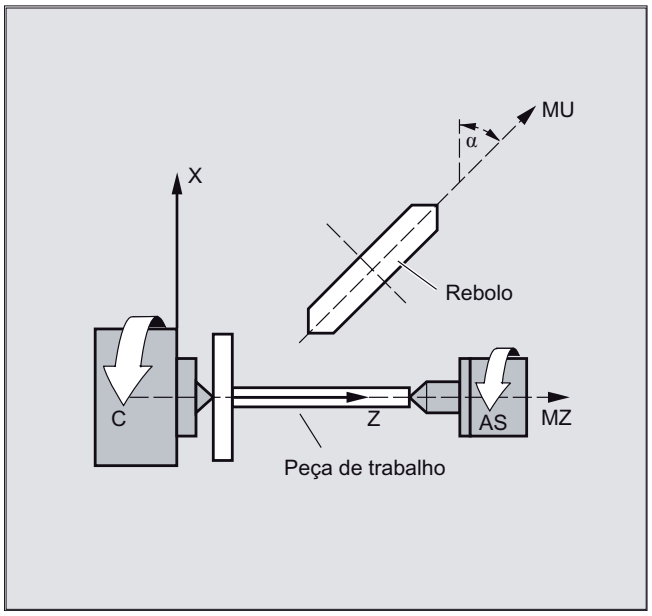
Com TRAFO_TYPE_n = 513 se programa a metade da largura da ranhura em OFFN. O contorno é executado com correção do raio da ferramenta OFFN.

6.8.3 Eixo inclinado (TRAANG)

Função

A função do eixo inclinado foi idealizada apenas para a tecnologia de retificação e possibilita as seguintes tarefas:

- Usinagem com eixo de penetração inclinado
- Para a programação se pode utilizar um sistema de coordenadas cartesiano.
- O comando transforma os movimentos de deslocamento programados do sistema de coordenadas cartesiano em movimentos de deslocamento dos eixos reais da máquina (caso padrão): eixo de penetração inclinado.



Sintaxe

```
TRAANG(α) ou TRAANG(α, n)
TRAFOOF
```

Significado

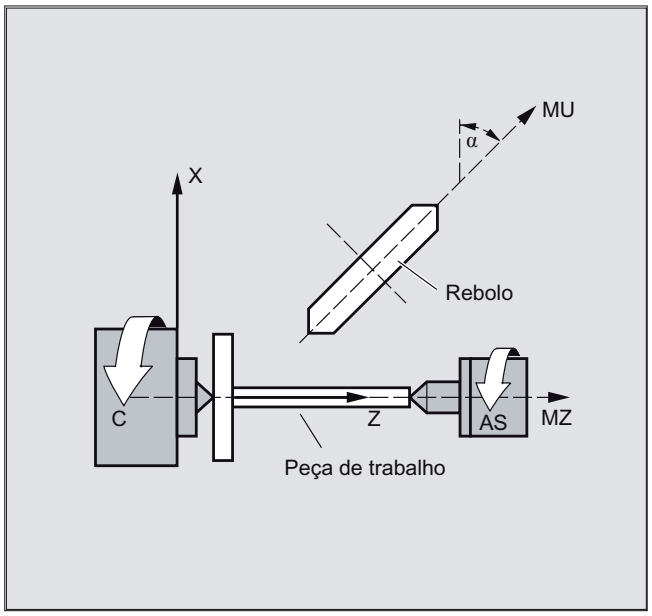
`TRAANG( ) ou TRAANG( ,n)`	Ativar a transformação com a parametrização da seleção anterior.
`TRAANG(α)`	Ativa a primeira transformação definida do eixo inclinado
`TRAANG(α,n)`	Ativa a transformação n. acordada do eixo inclinado. O n pode ser no máximo 2. TRAANG(α,1) corresponde ao TRAANG(α).
`αA`	Ângulo do eixo que está inclinado Os valores permitidos para α são: -90 graus< α < + 90 graus
`TRAFOOF`	Transformação desativada
`n`	Número de transformações acordadas

Supressão do ângulo α ou zero

Se o ângulo α for suprimido (p. ex. `TRAANG()`, `TRAANG(, n)`), será ativada a transformação com a parametrização da seleção anterior. Na primeira ativação se aplica a ocupação prévia conforme os dados da máquina.

Um ângulo α = 0 (p. ex. `TRAANG(0)`, `TRAANG(0,n)`) é uma parametrização válida e não corresponde mais à supressão do parâmetro em versões mais antigas.

Exemplo



Código de programa	**Comentário**
`N10 G0 G90 Z0 MU=10 G54 F5000 ->`	`; Seleção de ferramenta, compensação de`
`-> G18 G64 T1 D1`	`fixação, Seleção de plano`
`N20 TRAANG(45)`	`; Ativação da transformação do eixo inclinado`
`N30 G0 Z10 X5`	`; Aproximação da posição de saída`
`N40 WAITP(Z)`	`; Liberação de eixos para oscilação (vaivém)`
`N50 OSP[Z]=10 OSP2[Z]=5 OST1[Z]=-2 ->`	`; Oscilação, até a dimensão ser alcançada`
`-> OST2[Z]=-2 FA[Z]=5000`	`(Para oscilação, veja o capítulo`
`N60 OS[Z]=1`	`"Oscilação")`
`N70 POS[X]=4.5 FA[X]=50`	
`N80 OS[Z]=0`	

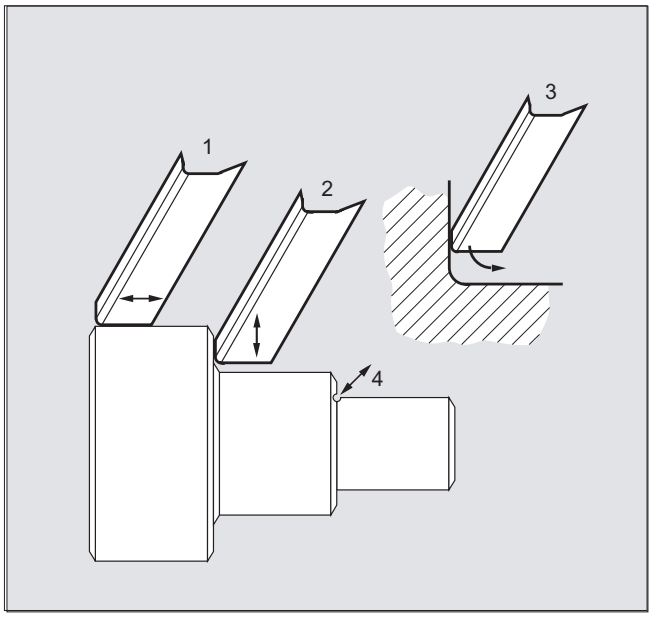
Código de programa	Comentário
N90 WAITP(Z)	; Liberação dos eixos oscilantes como eixos de posicionamento
N100 TRAFOOF	; Desativação da transformação
N110 G0 Z10 MU=10	; Afastamento
N120 M30	;

-> programar em um bloco

Descrição

Estão disponíveis as seguintes operações:

1. Retificação longitudinal
2. Retificação transversal
3. Retificação de um determinado contorno
4. Retificação inclinada de canal



Fabricante da máquina

Os seguintes ajustes são definidos através de dados de máquina:

- o ângulo entre um eixo de máquina e o eixo inclinado,
- a posição do ponto zero da ferramenta relativa à origem do sistema de coordenadas acordado na função "Eixo inclinado",
- a reserva de velocidade que é mantida de prontidão no eixo paralelo para o movimento de compensação,
- a reserva de aceleração que é mantida de prontidão no eixo paralelo para a aceleração de compensação.

Configuração de eixo

Para programar no sistema cartesiano de coordenadas, o comando deve informar a relação entre este sistema de coordenadas e os eixos realmente existentes da máquina (MU, MZ):

- Denominação dos eixos geométricos
- Atribuição dos eixos geométricos para eixos de canal
  - caso geral (eixo inclinado não ativo)
  - Eixo inclinado ativo
- Atribuição dos eixos de canal com os números de eixo da máquina
- Identificação dos fusos
- Atribuição de nomes de eixos de máquina

Com exceção do "Eixo inclinado ativo" o procedimento corresponde ao procedimento da configuração normal de eixos.

6.8.4 Programar eixo inclinado (G05, G07)

Função

Em modo JOG o rebolo pode ser movimentado de modo cartesiano ou no sentido do eixo inclinado (a exibição permanece cartesiana). Apenas se movimenta o eixo U real, a indicação do eixo Z é atualizada.

Os deslocamentos REPOS devem ser retornados em modo cartesiano.

A ultrapassagem dos limites cartesianos da área de trabalho são monitorados em modo JOG
com o "Deslocamento PTP" ativo, o eixo correspondente é freado antes. Se o "Deslocamento PTP" não estiver ativo, o eixo pode ser deslocado exatamente até o limite da área de trabalho.

Literatura

/FB2/ Manual de funções ampliadas; Transformação cinemática (M1)

Sintaxe

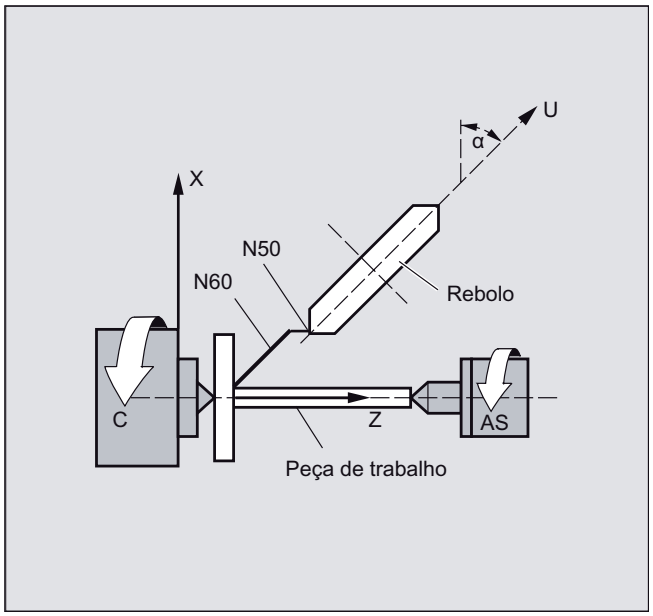
```
G07
G05
```

Os comandos `G07/G05` servem para facilitar a programação do eixo inclinado. Aqui se pode programar e exibir posições no sistema de coordenadas cartesiano. A correção da ferramenta e o deslocamento do ponto zero são incluídos de modo cartesiano. Após a programação do ângulo do eixo inclinado no programa NC, a posição de partida pode ser aproximada (`G07`) e, em seguida, ser executada usinagem do canal inclinado (`G05`).

Significado

G07	Aproximar posição de partida
G05	Ativa a usinagem inclinada de canais

Exemplo




```
Programação                                  Comentário
N.. G18                    ; Programar o ângulo do eixo inclinado
N50 G07 X70 Z40 F4000      ; Aproximar posição de partida
N60 G05 X70 F100           ; Executar usinagem de canal inclinado
N70 ...                    ;
```

6.9 Deslocamento PTP cartesiano

Função

Com esta função se pode programar uma posição em um sistema de coordenadas cartesiano, mas o movimento da máquina é realizado em coordenadas da máquina. Por exemplo, a função pode ser aplicada na mudança da posição da articulação se o movimento também for realizado através de uma singularidade.

Indicação

A função apenas tem relevância em conjunto com uma transformação ativa. Além disso o "Deslocamento PTP" somente é permitido com G0 e G1.

Sintaxe

```
N... TRAORI
N... STAT='B10' TU='B100' PTP
N... CP
```

Deslocamento PTP com transformação genérica de 5/6 eixos

Se na transformação genérica de 5/6 eixos ativa for ativado um deslocamento ponto-a-ponto com PTP no sistema de coordenadas `(ORIMKS)`, então a orientação da ferramenta pode ser programada tanto com posições de eixo rotativo

```
N... G1 X Y Z A B C
```

como com vetores de ângulo euleriano e ângulo RPY independentes da cinemática

```
N... ORIEULER ou ORIRPY
N... G1 X Y Z A2 B2 C2
```

ou com os vetores de direção

```
N... G1 X Y Z A3 B3 C3
```

. Neste caso pode estar ativa tanto uma interpolação de eixo rotativo como interpolação de vetor com interpolação de grande circunferência `ORIVECT` ou interpolação do vetor de orientação em uma superfície periférica cônica `ORICONxx`.

Ambigüidades da orientação com vetores

Na programação da orientação com vetores existe uma ambigüidade nas possíveis posições de eixo rotativo. Aqui as posições de eixo rotativo a serem aproximadas podem ser selecionadas através da programação do `STAT = <...>`. Se

for programado `STAT = 0` (corresponde ao ajuste padrão),
as posições serão aproximadas pela distância mais curta até as posições de partida. Se

for programado `STAT = 1`,
as posições serão aproximadas pela distância mais longa até as posições de partida.

Significado

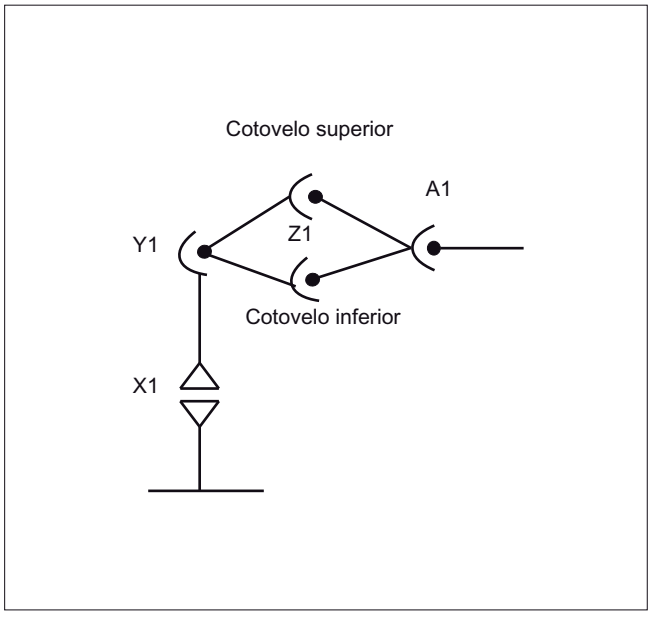
Os comandos `PTP` e `CP` são ativados modalmente. O ajuste padrão é o `CP`.

Enquanto a programação do valor STAT for aplicada como modal, a programação do TU = <...> atua por bloco.

Uma outra diferença é que a programação de um valor STAT apenas tem efeito na interpolação de vetor, enquanto que a programação do TU também é avaliada com a interpolação de eixo rotativo ativa.

`PTP`	**P**oint **t**o **P**oint (movimento ponto-a-ponto) O movimento é executado como movimento de eixo sincronizado; o eixo mais lento envolvido no movimento é o eixo dominante da velocidade.
`CP`	**c**ontinuous **p**ath (movimento de percurso) O movimento é executado como movimento de percurso cartesiano.
`STAT=`	Posição das articulações; o valor está em função da transformação.
`TU=`	A informação TURN é ativada por bloco. Com isso é possível aproximar claramente o ângulo de eixo entre -360 graus e +360 graus.

Exemplo




```
N10 G0 X0 Y-30 Z60 A-30 F10000
```
Posição de saída
→ Cotovelo superior

```
N20 TRAORI(1)
```
Transformação ativada

```
N30 X1000 Y0 Z400 A0
```

```
N40 X1000 Z500 A0 STAT='B10'
TU='B100' PTP
```
Reorientação sem transformação
→ Cotovelo inferior

```
N50 X1200 Z400 CP                    Transformação reativada
N60 X1000 Z500 A20
N70 M30
```

Exemplo de deslocamento PTP na transformação genérica de 5 eixos

Suposição: Existe uma cinemática ortogonal CA como base.

Código de programa	Comentário
TRAORI	; Transformação de cinemática CA ativa
PTP	; Ativar deslocamento PTP
N10 A3 = 0 B3 = 0 C3 = 1	; Posições de eixo rotativo C = 0 A = 0
N20 A3 = 1 B3 = 0 C3 = 1	; Posições de eixo rotativo C = 90 A = 45
N30 A3 = 1 B3 = 0 C3 = 0	; Posições de eixo rotativo C = 90 A = 90
N40 A3 = 1 B3 = 0 C3 = 1 STAT = 1	; Posições de eixo rotativo C = 270 A = –45

Selecionar uma posição de aproximação única da posição de eixo rotativo:

No bloco N40, através da programação do `STAT = 1`, os eixos rotativos percorrem o curso mais longo de seu ponto de partida (C=90, A=90) ao ponto final (C=270, A=–45), ao contrário do `STAT = 0` onde se percorre o curso mais curto até o ponto final (C=90, A=45).

Descrição

Uma mudança entre o deslocamento cartesiano e o deslocamento dos eixos de máquina é realizada através dos comandos `PTP` e `CP`.

Deslocamento PTP com transformação genérica de 5/6 eixos

No deslocamento PTP, ao contrário da transformação de 5/6 eixos, o TCP geralmente não permanece fixo no local, se apenas for alterada a orientação. As posições finais transformadas de todos os eixos da transformação (3 eixos lineares e até 3 eixos rotativos) são aproximadas linearmente, sem que a transformação ainda esteja ativa no próprio sentido.

O deslocamento PTP é desativado através da programação do código G modal `CP`.

As diversas transformações estão contidas na publicação:
/FB3/ Manual de funções especiais; Pacote de transformação Handling (TE4).

Programação da posição (STAT=)

Não se define claramente uma posição da máquina apenas com a especificação da posição com coordenadas cartesianas e a orientação da ferramenta. Dependendo da cinemática tratada, podem existir até 8 posições de articulação diferentes. Com isso estas são específicas de transformação. Para converter claramente uma posição cartesiana para o ângulo do eixo, se deve especificar a posição da articulação com o comando `STAT=` . O comando "`STAT`" contém um Bit como valor binário para cada uma das possíveis posições.

Bits de posição que devem ser programados com "`STAT`", veja:
/FB2/ Manual de funções ampliadas; Transformação cinemática (M1), Capítulo "Deslocamento PTP cartesiano".

Programação do ângulo de eixo (TU=)

Para aproximar claramente o ângulo de eixo < ±360 graus, esta informação deve ser programada com o comando "TU=".

Os eixos deslocam-se pelo curso mais curto:

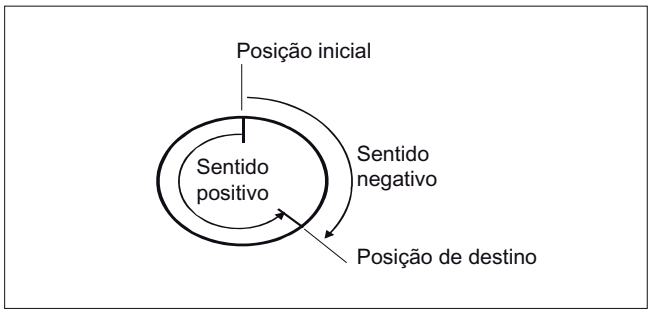
- se não foi programado nenhum TU para uma posição,
- em eixos que possuem uma área de deslocamento > ±360 graus.

Exemplo:

A posição de destino indicada na figura pode ser aproximada em sentido negativo ou positivo. O sentido é programado no endereço A1.

A1=225°, TU=Bit 0, → sentido positivo

A1=−135°, TU=Bit 1, → sentido negativo



Exemplo de avaliação do TU para transformação genérica de 5/6 eixos e posições de destino

A variável TU contém, para cada eixo que participa da transformação, um Bit que indica o sentido de deslocamento. A atribuição dos Bits TU corresponde ao panorama de eixos de canal dos eixos rotativos. A informação TU somente é avaliada em até 3 eixos rotativos que participam da transformação:

Bit0: Eixo 1, TU–Bit = 0 : 0 grau <= ângulo de eixo rotativo < 360 graus

Bit1: Eixo 2, TU–Bit = 1 : –360 graus < ângulo de eixo rotativo < 0 grau

A posição de partida de um eixo rotativo é C = 0, e através da programação de C = 270 o eixo rotativo se desloca até as seguintes posições de destino:

C = 270: TU–Bit 0, sentido de giro positivo

C = –90: TU–Bit 1, sentido de giro negativo

Outro comportamento

Mudança do modo de operação

A função "Deslocamento cartesiano PTP" apenas tem significado nos modos de operação AUTO e MDA. O atual ajuste é mantido na mudança do modo de operação para JOG.

Se estiver ajustado o código G `PTP`, os eixos serão movimentados no MCS. Se estiver ajustado o código G `CP`, os eixos serão movimentados no WCS.

Power On/RESET

Após o Power On ou após o RESET o ajuste estará em função do dado de máquina `$MC_GCODE_REST_VALUES[48]`. Como padrão está ajustado o tipo de deslocamento "`CP`".

REPOS

Se durante o bloco de interrupção estava ajustada a função "Deslocamento cartesiano PTP", então o reposicionamento também será realizado com `PTP`.

Movimentos sobrepostos

O deslocamento DRF ou o deslocamento externo de ponto zero são possíveis com restrições no deslocamento cartesiano PTP. Na mudança de um movimento PTP para um movimento CP não pode existir nenhuma sobreposição no BCS.

Suavização entre os movimentos CP e PTP

Com `G641` torna-se possível uma suavização programável de transição entre os blocos.

O tamanho da área de suavização é o percurso em mm ou polegadas que será suavizado na transição do bloco. O tamanho deve ser especificado como segue:

- para blocos G0 com `ADISPOS`
- para todos outros comandos de curso com `ADIS`

O processamento do percurso corresponde à consideração dos endereços F em blocos que não forem G0. O avanço é mantido nos eixos especificados no `FGROUP(...)`.

Cálculo de avanço

Para blocos CP são utilizados no cálculo os eixos cartesianos do sistema de coordenadas básico.

Para blocos PTP são utilizados no cálculo os eixos correspondentes do sistema de coordenadas da máquina.

6.9.1 PTP com TRANSMIT

Função

Com PTP no TRANSMIT se pode executar blocos G0 e G1 com otimização de tempo. Ao invés de deslocar linearmente os eixos do sistema de coordenadas básico (CP), são deslocados linearmente os eixos da máquina (PTP). Com isso o movimento do eixo da máquina atua próximo ao pólo de modo que o ponto final do bloco seja alcançado bem mais rápido.

O programa de peça continua sendo gravado no sistema de coordenadas cartesianas da peça e todos deslocamentos de coordenadas, rotações e programações de Frame continuam valendo. A simulação na HMI também é mostrada em sistema de coordenadas cartesianas da peça.

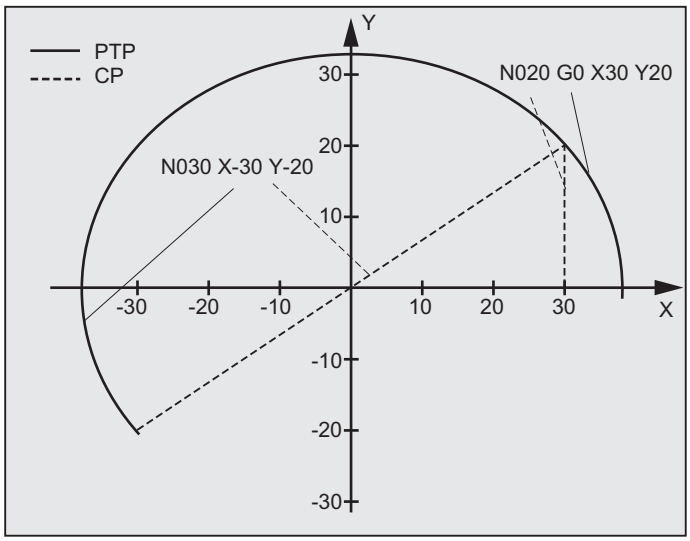
Sintaxe

```
N... TRANSMIT
N... PTPG0
N... G0 ...
...
N... G1 ...
```

Significado

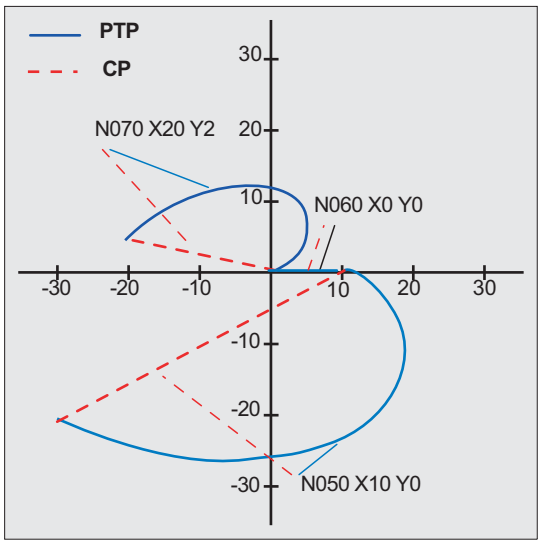
`TRANSMIT`	Ativa a primeira função TRANSMIT acordada (veja o capítulo "Operações de fresamento em peças torneadas: TRANSMIT")
`PTPG0`	**P**oint **t**o **P**oint **G0** (movimento ponto-a-ponto automático em cada bloco G0 e em seguida, passa novamente para CP) Visto que STAT e TU são modais, sempre se aplica o último valor programado.
`PTP`	**P**oint **t**o **P**oint (movimento ponto-a-ponto) Para o TRANSMIT o PTP significa que o afastamento em sistema cartesiano de espirais de arquimedes é realizado em torno do pólo ou a partir do pólo. Os movimentos de ferramenta resultantes disso ocorrem claramente diferente como no CP e estão representados nos respectivos exemplos de programação.
`STAT=`	Cancelamento da ambigüidade referente ao pólo.
`TU=`	O TU não é relevante no PTP com TRANSMIT

Exemplo de contorneamento do pólo com PTP e TRANSMIT



Código de programa	**Comentário**
N001 G0 X30 Z0 F10000 T1 D1 G90	; Posição de saída da dimensão absoluta
N002 SPOS=0	
N003 TRANSMIT	; Transformação TRANSMIT
N010 PTPG0	; para cada bloco G0 automaticamente PTP e depois novamente CP
N020 G0 X30 Y20	
N030 X-30 Y-20	
N120 G1 X30 Y20	
N110 X30 Y0	
M30	

Exemplo de afastamento do pólo com PTP e TRANSMIT



Programação	Comentário
N001 G0 X90 Z0 F10000 T1 D1 G90	; Posição de saída
N002 SPOS=0	
N003 TRANSMIT	; Transformação TRANSMIT
N010 PTPG0	; para cada bloco G0 automaticamente PTP e depois novamente CP
N020 G0 X90 Y60	
N030 X-90 Y-60	
N040 X-30 Y-20	
N050 X10 Y0	
N060 X0 Y0	
N070 X-20 Y2	
N170 G1 X0 Y0	
N160 X10 Y0	
N150 X-30 Y-20	
M30	

Descrição

PTP e PTPG0

O PTPG0 é considerado em todas transformações que podem processar o PTP. Nos demais casos o PTPG0é irrelevante.

Os blocos G0 são executados em modo CP.

A seleção do PTP e PTPG0 é realizada no programa de peça ou através da desseleção do CP no dado de máquina $MC_GCODE_RESET_VALUES[48].

⚠ CUIDADO

Condições gerais

Em relação aos movimentos da ferramenta e a colisão estão valendo várias condições gerais e determinadas exceções de função como:

Com PTP não pode existir nenhuma correção de raio de ferramenta (WRK) ativa.

Com PTPG0 o deslocamento é realizado pelo CP com a correção de raio de ferramenta (WRK) ativada.

Com PTP não é possível realizar uma aproximação e afastamento suave (WAB).

Com PTPG0 o deslocamento é realizado pelo CP com a aproximação e afastamento suave (WAB).

Com PTP não é possível realizar ciclos de desbaste (CONTPRON, CONTDCON).

Com PTPG0 o deslocamento é realizado com CP nos ciclos de desbaste (CONTPRON, CONTDCON).

Chanfros (CHF, CHR) e arredondamentos (RND, RNDM) são ignorados.

O compressor não é compatível com o PTP e sua seleção é automaticamente desfeita nos blocos PTP.

Uma sobreposição de eixos na interpolação não pode ser alterada durante o segmento PTP.

Com G643 passa-se automaticamente para suavização com precisão axial G642.

Com o PTP ativo os eixos da transformação não podem ser eixos de posicionamento ao mesmo tempo.

Literatura:

/FB2/ Manual de funções ampliadas; Transformação cinemática (M1),
Capítulo "Deslocamento PTP cartesiano"

PTP com TRACON:

O PTP também pode ser utilizado com o TRACON, se a primeira transformação PTP encadeada oferecer o suporte.

Significado do STAT= e TU= com TRANSMIT

Se o eixo rotativo deve girar 180 graus e o contorno deve passar pelo pólo em CP, os eixos rotativos podem girados em função do dado de máquina $MC_TRANSMIT_POLE_SIDE_FIX_1/2 [48] em -/+ 180 graus e deslocados no sentido horário ou anti-horário. Da mesma forma, se pode ajustar para que seja realizado o deslocamento ou a rotação em torno do pólo.

6.10 Condições gerais na seleção de uma transformação

Função

A seleção das transformações é possível através do programa de peça ou MDA. Aqui se deve observar:

- Não se insere um bloco intermediário de movimento (chanfros/raios).
- Uma sucessão de blocos Spline deve estar concluída, caso contrário aparece uma mensagem.
- A correção fina de ferramenta deve estar desselecionada (FTOCOF); caso contrário aparece uma mensagem.
- A correção de raio da ferramenta deve estar desselecionada (G40); caso contrário aparece uma mensagem.
- Um correção ativa do comprimento de ferramenta é assumida pelo comando na transformação.
- O atual Frame ativo antes da transformação é desselecionado pelo comando.
- Um limite ativo de área de trabalho é desselecionado pelo comando nos eixos afetados pela transformação (equivale ao WALIMOF).
- A monitoração da área de proteção é desselecionada.
- O modo de controle da trajetória e a suavização são cancelados.
- Todos os eixos especificados no dado de máquina devem ser sincronizados relativos ao bloco.
- Eixos trocados são destrocados; caso contrário aparece uma mensagem.
- É emitida uma mensagem para os eixos dependentes.

Troca de ferramentas

Uma troca de ferramentas somente é permitida com a correção de raio da ferramenta desselecionada.

Uma troca da correção do comprimento da ferramenta e uma seleção/desseleção da correção do raio da ferramenta não podem ser programadas no mesmo bloco.

Mudança de Frames

São permitidas todas instruções relativas ao sistema de coordenadas básico (FRAME, correção do raio da ferramenta). Porém, uma mudança de Frames com G91 (dimensão incremental) não é tratada de modo especial, diferente como na transformação inativa. O incremento a ser executado é avaliado no sistema de coordenadas da peça de trabalho do novo Frame, independente de qual Frame estava estiver no bloco anterior.

Exceções

Os eixos afetados pela transformação não podem ser utilizados:

- como eixo Preset (alarme),
- para a aproximação do ponto fixo (alarme),
- para o referenciamento (alarme).

6.11 Desselecionar a transformação (TRAFOOF)

Função

Com o comando `TRAFOOF` são desativadas todas transformações e Frames ativos.

Indicação

Em seguida, os Frames necessários devem ser ativados através de uma nova programação.

Aqui se deve observar:

Para a desseleção da transformação são aplicadas as mesmas condições gerais como na seleção (veja o capítulo "Condições gerais na seleção de uma transformação").

Sintaxe

```
TRAFOOF
```

Significado

`TRAFOOF`	Comando para desativar todas as transformações e Frames ativos

6.12 Transformações encadeadas (TRACON, TRAFOOF)

Função

Sempre se pode ligar (encadear) **duas** transformações consecutivas, de modo que as partes de movimento dos eixos da primeira transformação se tornam os dados de entrada para a segunda transformação, a encadeada. As partes de movimento da segunda transformação atuam nos eixos da máquina.

A cadeia pode compreender **duas** transformações.

Indicação

Uma ferramenta sempre é atribuída à primeira transformação de uma cadeia. A transformação subsequente tem um comportamento como se o comprimento ativo da ferramenta fosse zero. Apenas estão ativos os comprimentos básicos da ferramenta (_BASE_TOOL_) ajustados através de dados de máquina para a primeira transformação da cadeia.

Fabricante da máquina

Observe as informações do fabricante da máquina referente às eventuais transformações predefinidas através de dados de máquina.

As transformações e transformações encadeadas são opcionais. O atual catálogo fornece informações sobre a disponibilidade de determinadas transformações da cadeia em determinados comandos.

Aplicações

- Retificação de contornos que foram programados como linha periférica de um desenvolvimento de cilindro (TRACYL), com um rebolo de retificação inclinado, p. ex. retificação de ferramenta.
- Usinagem fina de um contorno não redondo produzido com TRANSMIT e rebolo de retificação inclinado.

Sintaxe

`TRACON(trf,par)`	Se ativa uma transformação encadeada.
`TRAFOOF`	

Significado

TRACON	A transformação encadeada é ativada. Outra transformação ativada antes é desativada de modo implícito pelo TRACON().
TRAFOOF	A última transformação ativada (encadeada) é desativada.
trf	Número da transformação encadeada: 0 ou 1 para primeira/única transformação encadeada. Se na primeira casa não foi programado nada, isto significa o mesmo que a especificação do valor 0 ou 1, ou seja, se ativa a primeira/única transformação. 2 para a segunda transformação encadeada. (valores diferentes de 0 - 2 geram um alarme de erro).
par	Um ou mais parâmetros separados por vírgula para as transformações no encadeamento que requerem parâmetros, como o ângulo do eixo inclinado. No caso de parâmetros não especificados serão ativados os pré-ajustes ou os parâmetros que foram usados por último. Com a colocação de vírgulas deve-se providenciar para que os parâmetros sejam avaliados na sequência em que forem solicitados, se os pré-ajustes devem atuar nos parâmetros precedentes. Especialmente na especificação deve haver pelo menos um parâmetro antes desta vírgula, mesmo se a especificação de trf não for necessária, como o TRACON(, 3.7).

Requisitos

A **segunda** transformação deve se um **"Eixo inclinado"** (TRAANG). Como primeira transformação são possíveis:

- Transformações de orientação (TRAORI), inclusive cabeçote de fresar cardânico
- TRANSMIT
- TRACYL
- TRAANG

Para usar o comando de ativação de uma transformação encadeada é necessário que cada transformação a ser encadeada e a transformação encadeada a ser ativada esteja definida através de dados de máquina.

O uso de condições gerais e casos especiais especificados nas descrições individuais das transformações também deve ser observado em um encadeamento.

Informações sobre projeção dos dados de máquina das transformações estão disponíveis no(a):

/FB2/ Manual de funções ampliadas; Transformações cinemáticas (M1) e

/FB3/ Manual de funções especiais; Transformações de 3 a 5 eixos (F2).

Corretores de ferramentas 7

7.1 Memória de correções

Função

Estrutura da memória de correções

Cada campo de dados pode ser chamado com um número T ou D (exceto "Nº D plano") e, além das informações geométricas da ferramenta, também contém outras especificações como por exemplo o tipo de ferramenta.

Estrutura plana de números D

A "estrutura plana de números D" é utilizada se o gerenciamento de ferramentas for realizado fora da NCK. Neste caso os números D são criados com os respectivos blocos de correção de ferramenta sem a atribuição às ferramentas.

No programa de peça ainda se pode programar o T. Mas este T não tem relação com o número D programado.

Dados de corte do usuário

Através de dados de máquina podem ser configurados dados de corte do usuário. Observe as instruções do fabricante da máquina.

Parâmetro da ferramenta

Indicação

Valores individuais na memória de correções

Os diversos valores da memória de correções P1 até P25 podem ser lidos e gravados no programa através de variáveis de sistema. Os demais parâmetros estão reservados.

Os parâmetros de ferramenta $TC_DP6 até $TC_DP8, $TC_DP10 e $TC_DP11, $TC_DP15 até $TC_DP17, $TC_DP19 e $TC_DP20 têm outro significado, independentemente do tipo de ferramenta.

[1]No caso das fresas também é aplicado no fresamento de topo 3D
[2]Reservado para serra de ranhuras tipo
[3]: Não é utilizado pelo SINUMERIK 840D

Parâmetro de ferramenta Número (DP)	Significado das variáveis de sistema	Observação
$TC_DP1	Tipo de ferramenta	Como vista geral veja a lista
$TC_DP2	Posição de corte	apenas para ferramentas de tornear
Geometria	**Correção de comprimentos**	
$TC_DP3	Comprimento 1	Processamento por
$TC_DP4	Comprimento 2	Tipo e plano
$TC_DP5	Comprimento 3	
Geometria	**Raio**	
$TC_DP6[1] $TC_DP6[2]	Raio 1 / comprimento 1 diâmetro d	Ferram. de fresar/tornear/retificar Serra para ranhuras
$TC_DP7[1] $TC_DP7[2]	Comprimento 2 / raio de canto de fresa cônica Largura de ranhura b raio de canto	Fresas Serra para ranhuras
$TC_DP8[1] $TC_DP8[2]	Raio de arredondamento 1 para ferramentas de fresar Ressalto k	Fresas Serra para ranhuras
$TC_DP9[1,3]	Raio de arredondamento 2	Reservado
$TC_DP10[1]	Ângulo 1 lado frontal da ferramenta	fresas cônicas
$TC_DP11[1]	Ângulo 2 eixo longitudinal da ferramenta	fresas cônicas
Desgaste	**Correção de comprimento e raio**	
$TC_DP12	Comprimento 1	
$TC_DP13	Comprimento 2	
$TC_DP14	Comprimento 3	
$TC_DP15[1] $TC_DP15[2]	Raio 1 / comprimento 1 diâmetro d	Ferramenta de fresar/tornear/retificar Serra para ranhuras
$TC_DP16[1] $TC_DP16[3]	Comprimento 2 / Raio de canto de fresa cônica Largura de ranhura b raio de canto	Fresas Serra para ranhuras
$TC_DP17[1] $TC_DP17[2]	Raio de arredondamento 1 para fresas Ressalto k	Fresas / fresamento de topo 3D Serra para ranhuras
$TC_DP18[1,3]	Raio de arredondamento 2	Reservado
$TC_DP19[1]	Ângulo 1 lado frontal da ferramenta	fresas cônicas
$TC_DP20[1]	Ângulo 2 eixo longitudinal da ferramenta	fresas cônicas
Medida básica/Adaptador	**Correções dos comprimentos**	
$TC_DP21	Comprimento 1	
$TC_DP22	Comprimento 2	
$TC_DP23	Comprimento 3	
Tecnologia		
$TC_DP24	Ângulo livre	apenas para ferramentas de tornear
$TC_DP25		Reservado

Observações

Para as grandezas geométricas (p. ex. comprimento 1 ou raio) existem vários componentes de entrada. Estes são processados aditivamente para uma grandeza resultante (p. ex. comprimento total 1, raio total), que entrará em ação.

As correções desnecessárias devem ser especificadas com o valor zero.

Parâmetros de ferramenta $TC-DP1 até $TC-DP23 com ferramentas de contorno

Indicação

Os parâmetros de ferramenta que não estão mencionados na tabela, como o $TC_DP7, não são avaliados, isto é, seu conteúdo não tem relevância.

Parâmetro de ferramenta Número (DP)	Significado	Cortes Dn		Observação
$TC_DP1	Tipo de ferramenta			400 até 599
$TC_DP2	Posição de corte			
Geometria	**Correção de comprimentos**			
$TC_DP3	Comprimento 1			
$TC_DP4	Comprimento 2			
$TC_DP5	Comprimento 3			
Geometria	**Raio**			
$TC_DP6	Raio			
Geometria	**Ângulo limite**			
$TC_DP10	Ângulo limite mínimo			
$TC_DP11	Ângulo limite máximo			
Desgaste	**Correção de comprimento e raio**			
$TC_DP12	Desgaste de comprimento 1			
$TC_DP13	Desgaste de comprimento 2			
$TC_DP14	Desgaste de comprimento 3			
$TC_DP15	Desgaste do raio			
Desgaste	**Ângulo limite**			
$TC_DP19	Ângulo limite mín. de desgaste			
$TC_DP20	Ângulo limite máx. de desgaste			
Medida básica/Adaptador	**Correções dos comprimentos**			
$TC_DP21	Comprimento 1			
$TC_DP22	Comprimento 2			
$TC_DP23	Comprimento 3			

Valor básico e valor de desgaste

Os tamanhos resultantes sempre resultam da soma do valor básico com o valor de desgaste (p. ex. $TC_DP6 + $TC_DP15 para o raio). Para o comprimento da ferramenta ainda é adicionada a dimensão básica ($TC_DP21 – $TC_DP23) do primeiro corte. Neste comprimento de ferramenta também terão efeito todas as demais grandezas que podem influenciar o comprimento efetivo de uma ferramenta convencional (como adaptadores, porta-ferramenta orientável, dados de ajuste).

Ângulo limite 1 e 2

Os ângulos limite 1 e 2 sempre têm sua referência no vetor do centro do corte ao ponto de referência do corte e são contados em sentido anti-horário.

7.2 Corretores aditivos

7.2.1 Ativar correções aditivas (DL)

Função

As correções aditivas podem ser consideradas como correções programáveis do processo de usinagem. Elas têm referência nos dados geométricos de um corte com isso são parte integrante dos dados de corte da ferramenta.

Os dados de uma correção aditiva são solicitados através de um número DL (DL:Location dependent; correções referentes ao respectivo local de aplicação) e especificados através da interface de operação.

Aplicação

Através das correções aditivas podem ser compensados erros de medição condicionados ao local de aplicação.

Sintaxe

```
DL=<número>
```

Significado

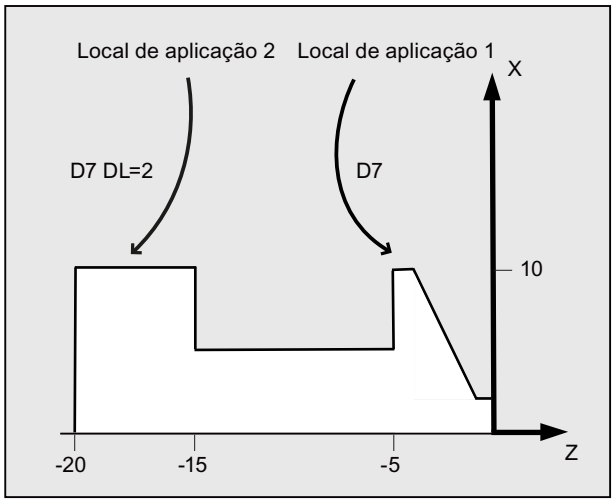
`DL`	Comando para ativação de uma correção aditiva
`<número>`	Através do parâmetro `<número>` é especificado o bloco aditivo de dados de correção da ferramenta a ser ativado.

Indicação

A definição da quantidade e ativação das correções aditivas são realizadas através de dados de máquina (→ Observar as informações do fabricante da máquina!).

Exemplo

O mesmo corte é utilizado para 2 assentos de rolamento:



Código de programa	Comentário
N110 T7 D7	; O revólver está posicionado no alojamento 7. D7 e DL=1 são ativados e executados no próximo bloco.
N120 G0 X10 Z1	
N130 G1 Z-6	
N140 G0 DL=2 Z-14	; De forma aditiva ao D7 se ativa o DL=2 e executado no próximo bloco.
N150 G1 Z-21	
N160 G0 X200 Z200	; Aproximar o ponto de troca de ferramentas.
...	

7.2.2 Definir valores de desgaste e de ajuste ($TC_SCPxy[t,d], $TC_ECPxy[t,d])

Função

Os valores de desgaste e de ajuste podem ser lidos e gravados através de variáveis de sistema. Aqui a lógica se orienta na lógica das respectivas variáveis de sistema para ferramentas e cortes.

Variáveis de sistema

Variável de sistema	Significado
$TC_SCPxy[<t>,<d>]	Valores de desgaste que estão atribuídos aos respectivos parâmetros geométricos através de xy, onde x corresponde ao número do valor de desgaste e y estabelece a relação com o parâmetro geométrico.
$TC_ECPxy[<t>,<d>]	Valores de ajuste que estão atribuídos aos respectivos parâmetros geométricos através de xy, onde x corresponde ao número do valor de ajuste e y estabelece a relação com o parâmetro geométrico.
<t>: Número T da ferramenta <d>: Número D do corte da ferramenta	

Indicação

Os valores definidos de desgaste e de ajuste são adicionados aos parâmetros geométricos e aos demais parâmetros de correção (número D).

Exemplo

O valor de desgaste do comprimento 1 é definido para o corte <d> da ferramenta <t> no valor 1.0.

Parâmetro: $TC_DP**3** (Comprimento 1, para ferramentas de tornear)

Valores de desgaste: $TC_SCP1**3** até $TC_SCP6**3**

Valores de ajuste: $TC_ECP1**3** até $TC_ECP6**3**

$TC_SCP4**3** [<t>,<d>] = 1.0

7.2.3 Deletar correções aditivas (DELDL)

Função

Com o comando DELDL são deletadas correções aditivas para o corte de uma ferramenta (liberação de memória). Aqui são deletados tanto os valores de desgaste definidos como os valores de ajuste definidos.

Sintaxe

```
DELDL[<t>,<d>]
DELDL[<t>]
DELDL
<estado>=DELDL[<t>,<d>]
```

Significado

DELDL	Comando para deletar correções aditivas
<t>	Número T da ferramenta
<d>	Número D do corte da ferramenta
DELDL[<t>,<d>]	São deletadas todas as correções aditivas do corte <d> da ferramenta <t>.
DELDL[<t>]	São deletadas todas as correções aditivas de todos os cortes da ferramenta <t>.
DELDL	São deletadas todas as correções aditivas de todos cortes de todas ferramentas da unidade TO (para o canal em que o comando for programado).
<estado>	Estado de deletação

Valor:	Significado:
0	A deletação foi realizada com sucesso.
-	A deletação não foi executada (quando a parametrização denomina exatamente um corte), ou a deletação não foi executada totalmente (quando a parametrização denomina vários cortes).

Indicação

Os valores de desgaste e de ajuste de ferramentas ativas não podem ser deletados (comportamento semelhante à deletação do D e de dados de ferramenta).

7.3 Tratamentos especiais de corretor de ferramenta

Função

Com os dados de ajuste SD42900 até SD42960 se pode controlar a avaliação do sinal prévio para comprimento de ferramenta e desgaste.

Isto também se aplica ao comportamento dos componentes de desgaste no espelhamento de eixos geométricos ou na mudança de plano de usinagem e ainda para compensação de temperatura no sentido da ferramenta.

Valores de desgaste

Se a seguir forem tomados como referência valores de desgaste, entendemos sempre como a soma dos próprios valores de desgaste ($TC_DP12 bis $TC_DP20) e das correções aditivas com os valores de desgaste ($SCPX3 até $SCPX11) e os valores de ajuste ($ECPX3 até $ECPX11).

Para mais informações sobre as correções aditivas, veja:
Literatura:
Manual de funções para gerenciamento de ferramentas

Dados de ajuste

Dados de ajuste	Significado
SD42900 $SC_MIRROR_TOOL_LENGTH	Espelhamento de componentes do comprimento de ferramentas e de componentes da dimensão básica.
SD42910 $SC_MIRROR_TOOL_WEAR	Espelhamento de valores de desgaste dos componentes do comprimento de ferramentas.
SD42920 $SC_WEAR_SIGN_CUTPOS	Avaliação do sinal precedente de componentes de desgaste em função da posição do corte.
SD42930 $SC_WEAR_SIGN	Inverte o sinal das dimensões de desgaste.
SD42935 $SC_WEAR_TRANSFORM	Transformação dos valores de desgaste.
SD42940 $SC_TOOL_LENGTH_CONST	Associação dos componentes de comprimento de ferramenta com os eixos geométricos.
SD42950 $SC_TOOL_LENGTH_TYPE	Associação dos componentes de comprimento de ferramenta independentemente do tipo de ferramenta.
SD42960 $SC_TOOL_TEMP_COMP	Valor de compensação de temperatura no sentido da ferramenta. Também está ativo com a orientação de ferramenta presente.

Literatura

Manual de funções básicas; Correção de ferramenta (W1)

Outras informações

Ativação dos dados de ajuste alterados

A nova avaliação de componentes de ferramentas em uma alteração dos dados de ajuste descritos somente se ativa na próxima vez que o corte de ferramenta for selecionado. Se uma ferramenta está ativa e a avaliação dos dados alterados desta ferramenta deve ser ativada, esta ferramenta deve ser selecionada novamente.

O mesmo se aplica no caso em que o comprimento resultante da ferramenta sofrer alteração, pois o estado do espelhamento de um eixo foi alterado. A ferramenta deve ser selecionada novamente depois do comando de espelhamento, para que os componentes alterados do comprimento da ferramenta possam ser ativados.

Porta-ferramenta orientável e novos dados de ajuste

Os dados de ajuste SD42900 até SD42940 não atuam sobre os componentes de um porta-ferramenta orientável que eventualmente esteja ativo. Entretanto, uma ferramenta com seu comprimento total resultante (comprimento da ferramenta + desgaste + dimensão básica) sempre entra no processamento com um porta-ferramenta orientável. No processamento do comprimento total resultante são consideradas todas alterações geradas através dos dados de ajuste, ou seja, os vetores do porta-ferramenta orientável independem do plano de usinagem.

Indicação

Freqüentemente, ao serem empregados porta-ferramentas orientáveis, será mais conveniente definir todas ferramentas para um sistema básico não espelhado, mesmo para aquelas que serão utilizadas em operações espelhadas. Na usinagem com eixos espelhados o porta-ferramenta será girado de modo que a posição real da ferramenta seja descrita de forma correta. Todos componentes de comprimento da ferramenta atuam automaticamente no sentido correto, de modo que para o comando apenas resta a avaliação de alguns componentes através de dados de ajuste em função do estado de espelhamento de determinados eixos.

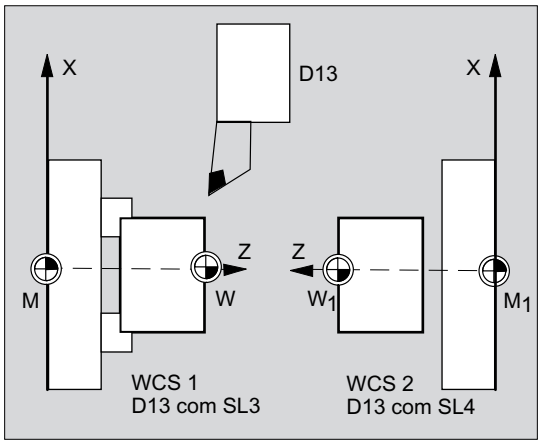
Outras possibilidades de aplicação

O uso da funcionalidade de porta-ferramentas orientáveis também pode ser bastante útil, se na máquina não houver nenhuma possibilidade física de girar as ferramentas, mas as ferramentas serem instaladas definitivamente com diferentes orientações. O dimensionamento da ferramenta pode ser realizado em uma orientação, e as dimensões relevantes à usinagem resultam através de rotações do porta-ferramenta virtual.

7.3.1 Espelhar comprimentos de ferramentas

Função

Com os dados de ajuste SD42900 $SC_MIRROR_TOOL_LENGTH e SD42910 $SC_MIRROR_TOOL_WEAR definidos diferente de zero, podemos espelhar componentes de comprimento de ferramentas e componentes da dimensão básica com valores de desgaste em seus respectivos eixos.



SD42900 $SC_MIRROR_TOOL_LENGTH

Dado de ajuste **diferente** de zero:

Os componentes de comprimento de ferramenta ($TC_DP3, $TC_DP4 e $TC_DP5) e os componentes da dimensão básica ($TC_DP21, $TC_DP22 e $TC_DP23), cujos eixos estão espelhados, também são espelhados através da inversão de sinais.

Os valores de desgaste **não** são espelhados juntos. Se estes também devem ser espelhados, então se deve empregar o dado de ajuste SD42910 $SC_MIRROR_TOOL_WEAR.

SD42910 $SC_MIRROR_TOOL_WEAR

Dado de ajuste **diferente** de zero:

Os valores de desgaste dos componentes de comprimento da ferramenta, cujos eixos estão espelhados, também são espelhados através da inversão de sinais.

7.3.2 Avaliação de sinais de desgaste

Função

Com os dados de ajuste SD42920 $SC_WEAR_SIGN_CUTPOS e SD42930 $SC_WEAR_SIGN definidos diferente de zero, podemos inverter a avaliação de sinais dos componentes de desgaste.

SD42920 $SC_WEAR_SIGN_CUTPOS

Dado de ajuste **diferente** de zero:

Nas ferramentas com posição de corte relevante (ferramentas de tornear e de retificar, tipos de ferramenta 400) a avaliação de sinais dos componentes de desgaste no plano de usinagem depende da posição do corte. Nos tipos de ferramenta sem posição de corte relevante, este dado de ajuste não tem nenhum significado.

Na seguinte tabela estão marcadas com um X as dimensões cujo sinal é invertido através do SD42920 (diferente de 0):

Posição de corte	Comprimento 1	Comprimento 2
1		
2		X
3	X	X
4	X	
5		
6		
7		X
8	X	
9		

Indicação

A avaliação de sinais através do SD42920 e do SD42910 é independente uma da outra. Por exemplo, se o sinal de uma dimensão se alterar através dos dois dados de ajuste, então o sinal resultante permanece inalterado.

SD42930 $SC_WEAR_SIGN

Dado de ajuste **diferente** de zero:

O sinal de todas dimensões de desgaste é invertido. O efeito ocorre tanto no comprimento da ferramenta como nas demais grandezas como raio de ferramenta, raio de arredondamento, etc.

Se for especificada uma dimensão de desgaste, então a ferramenta será "mais curta" e "mais estreita", veja o capítulo "Correção de ferramenta, tratamento especial", "Ativação dos dados de ajuste alterados".

7.3.3 Sistema de coordenadas da usinagem ativa (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS)

Função

Em função da cinemática da máquina ou da presença de um porta-ferramenta orientável, os valores de desgaste medidos em um destes sistemas de coordenadas serão transportados e transformados para um sistema de coordenadas adequado.

Sistemas de coordenadas da usinagem ativa

A partir dos seguintes sistemas de coordenadas podem resultar Offsets do comprimento da ferramenta, com os quais se inclui o componente desgaste de comprimento de ferramenta através do respectivo código G do grupo 56 em uma ferramenta ativa.

- Sistema de coordenadas da máquina (MCS)
- Sistema de coordenadas básico (BCS)
- Sistema de coordenadas da peça (WCS)
- Sistema de coordenadas da ferramenta (TCS)
- Sistema de coordenadas da ferramenta da transformação cinemática (KCS)

Sintaxe

```
TOWSTD
TOWMCS
TOWWCS
TOWBCS
TOWTCS
TOWKCS
```

Significado

`TOWSTD`	Valor de ajuste básico para correções no valor de desgaste do comprimento da ferramenta
`TOWMCS`	Correções no comprimento da ferramenta em MCS
`TOWWCS`	Correções no comprimento da ferramenta em WCS
`TOWBCS`	Correções no comprimento da ferramenta em BCS
`TOWTCS`	Correções do comprimento da ferramenta no ponto de referência do porta-ferramenta (porta-ferramenta orientável)
`TOWKCS`	Correções do comprimento da ferramenta do cabeçote de ferramenta (transformação cinemática)

Outras informações

Características de diferenciação

Na seguinte tabela estão representadas as características mais importantes de diferenciação:

Código G	Valor de desgaste	Porta-ferramenta orientável ativo
TOWSTD	Valor de ajuste básico, comprimento da ferramenta	Os valores de desgaste estão sujeitos à rotação.
TOWMCS	Valor de desgaste em MCS. O TOWMCS é idêntico ao TOWSTD, quando nenhum porta-ferramenta orientável estiver ativo.	Apenas se gira o vetor do comprimento de ferramenta resultante, sem consideração do desgaste.
TOWWCS	O valor de desgaste é convertido do WCS para MCS.	O vetor da ferramenta é processado sem consideração do desgaste, como no TOWMCS.
TOWBCS	O valor de desgaste é convertido do BCS para MCS.	O vetor da ferramenta é processado sem consideração do desgaste, como no TOWMCS.
TOWTCS	O valor de desgaste é convertido do sistema de coordenadas da ferramenta para o MCS.	O vetor da ferramenta é processado sem consideração do desgaste, como no TOWMCS.

TOWWCS , TOWBCS, TOWTCS: O vetor de desgaste é adicionado ao vetor da ferramenta.

Transformação linear

A definição do comprimento da ferramenta somente é conveniente em MCS se o MCS resultar do BCS através de uma transformação linear.

Transformação não linear

Se, por exemplo, uma transformação não linear estiver ativa com TRANSMIT, então se utiliza automaticamente o BCS quando se especifica o MCS como sistema de coordenadas desejado.

Nenhuma transformação cinemática e nenhum porta-ferramenta orientável

Se nem uma transformação cinemática nem um porta-ferramenta orientável estiver ativo, então, com exceção do WCS, os demais quatro sistemas de coordenadas coincidirão. Com isso apenas o WCS se diferencia dos demais. Visto que são avaliados somente os comprimentos de ferramentas, as transformações entre os sistemas de coordenadas não têm nenhuma importância.

Literatura:
Para mais informações sobre correções de ferramentas, veja:
Manual de funções básicas; Correções de ferramenta (W1)

Inclusão dos valores de desgaste

O dado de ajuste **SD42935 $SC_WEAR_TRANSFORM** define quais dos três componentes de desgaste:

- Desgaste
- Correções aditivas finas
- Correções aditivas aproximadas

serão submetidas à uma rotação através de uma transformação de adaptador ou de um porta-ferramenta orientável, se um dos seguintes códigos G estiver ativo:

- `TOWSTD` Posição inicial

  para correções no comprimento da ferramenta
- `TOWMCS` Valores de desgaste

  no sistema de coordenadas da máquina (MCS)
- `TOWWCS` Valores de desgaste

  no sistema de coordenadas da peça (WCS)
- `TOWBCS` Valores de desgaste (BCS)

  no sistema de coordenadas básico
- `TOWTCS` Valores de desgaste no sistema de coordenadas da ferramenta no assento do porta-ferramenta (referência do porta-ferramenta T)
- `TOWKCS` Valores de desgaste no sistema de coordenadas do cabeçote de ferramenta na transformação cinemática

---

Indicação

A avaliação dos diversos componentes de desgaste (atribuição aos eixos geométricos, avaliação de sinais) é influenciada através:

- do plano ativo
- da transformação de adaptador
- dos seguintes dados de ajuste:
  - SD42910 $SC_MIRROW_TOOL_WEAR
  - SD42920 $SC_WEAR_SIGN_CUTPOS
  - SD42930 $SC_WEAR_SIGN
  - SD42940 $SC_TOOL_LENGTH_CONST
  - SD42950 $SC_TOOL_LENGTH_TYPE

7.3.4 Comprimento de ferramenta e mudança de planos

Função

Com o dado de ajuste SD42940 $SC_TOOL_LENGTH_CONST definida diferente de zero, podemos atribuir os componentes de comprimento de ferramenta como comprimento, desgaste e dimensão básica aos eixos geométricos para ferramentas de tornear e retificar em uma mudança de planos.

SD42940 $SC_TOOL_LENGTH_CONST

Dado de ajuste **diferente** de zero:

A atribuição dos componentes de comprimento de ferramenta (comprimento, desgaste e dimensão básica) com os eixos geométricos não se altera na mudança do plano de usinagem (`G17` - `G19`).

A seguinte tabela mostra a associação dos componentes de comprimento de ferramenta com os eixos geométricos para ferramentas de tornear e de retificar (tipo de ferramenta 400 até 599):

Conteúdo	Comprimento 1	Comprimento 2	Comprimento 3
17	Y	X	Z
*)	X	Z	Y
19	Z	Y	X
-17	X	Y	Z
-18	Z	X	Y
-19	Y	Z	X

*) Cada valor diferente de 0, que não é igual a um dos seis valores mencionados, será avaliado como o valor 18.

A seguinte tabela mostra a associação dos componentes de comprimento de ferramenta com os eixos geométricos para todas demais ferramentas (tipo de ferramenta < 400 e > 599):

Plano de usinagem	Comprimento 1	Comprimento 2	Comprimento 3
*)	Z	Y	X
18	Y	X	Z
19	X	Z	Y
-17	Z	X	Y
-18	Y	Z	X
-19	X	Y	Z

*) Cada valor diferente de 0, que não é igual a um dos seis valores mencionados, será avaliado como o valor 17.

Indicação

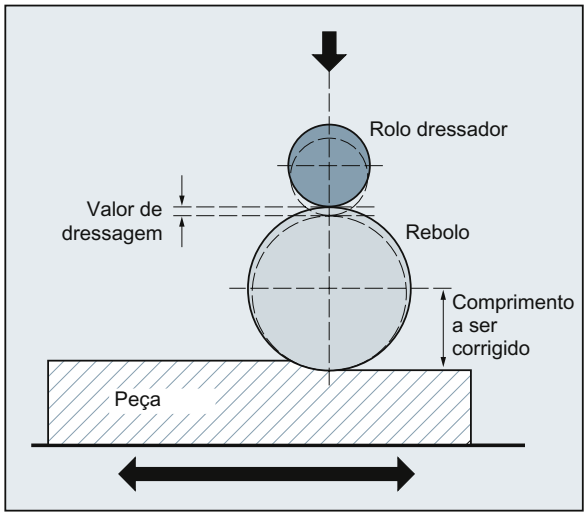
Na representação nas tabelas se pressupõe que os eixos geométricos até 3 sejam identificados com X, Y e Z. Determinante para a atribuição de uma correção para um eixo não é o identificador de eixo, mas a seqüência dos eixos.

7.4 Correção de ferramenta Online (PUTFTOCF, FCTDEF, PUTFTOC, FTOCON, FTOCOF)

Função

Com a função "Correção de ferramenta Online" ativa, no caso de ferramentas para retificação, é realizado um cálculo imediato da correção do comprimento de ferramenta que resulta da usinagem.

Um exemplo de aplicação é a dressagem, onde o rebolo é dressado paralelamente à usinagem:



A correção do comprimento da ferramenta pode ser alterada a partir do canal de usinagem ou de um canal paralelo (canal de dressagem).

Para gravar a correção de ferramenta Online são utilizadas funções diferentes dependendo do momento desejado da dressagem:

- Gravação contínua por bloco (`PUTFTOCF`)

  Com o `PUTFTOCF` a dressagem é executada simultaneamente com a usinagem.

  A correção da ferramenta é continuamente alterada no canal de usinagem conforme uma função de polinômio de 1º, 2º ou 3º grau, que deve ser definida primeiro com `FCTDEF`.

  O `PUTFTOCF` sempre atua por bloco, isto é, no bloco de deslocamento seguinte.

- Gravar continuamente de forma modal: `ID=1 DO FTOC` (veja "Correção Online da ferramenta (FTOC) (Página 605)")

- Gravação discreta (`PUTFTOC`)

  Com o `PUTFTOC` a dressagem não é executada simultaneamente com a usinagem a partir de um canal paralelo. O valor de correção especificado com `PUTFTOC` está imediatamente ativo no canal de destino.

Indicação

A correção de ferramenta Online somente pode ser aplicada em ferramentas para retificação.

Sintaxe

Desativação da correção de ferramenta Online no canal de destino:

```
FTOCON
...
FTOCOF
```

Gravação da correção de ferramenta Online:

- Continuamente por blocos:

```
FCTDEF(<função>,<LLimit>,<ULimit>,<a0>,<a1>,<a2>,<a3>)
PUTFTOCF(<função>,<valor de referência>,<parâmetro de ferramenta>,<canal>,<fuso>)
...
```

- Discreto:

```
PUTFTOC(<valor de correção>,<parâmetro de ferramenta>,<canal>,<fuso>)
...
```

Significado

`FTOCON`: Ativação da correção de ferramenta Online

O `FTOCON` deve ser programado no canal em que a correção de ferramenta Online será ativada.

`FTOCOF`: Cancelamento da correção de ferramenta Online

Com o `FTOCOF` a correção não é mais executada, mas nos dados de correção específicos de corte encontra-se o valor gravado totalmente corrigido com `PUTFTOC/PUTFTOCF`.

Nota:
Para a desativação definitiva da correção de ferramenta Online, após o `FTOCOF`, ainda precisa ocorrer uma ativação/desativação da ferramenta (`T...`).

`FCTDEF`: Com o `FCTDEF` é definida a função de polinômio para `PUTFTOCF`.

Parâmetro:

`<função>`: Número da função de polinômio
Tipo: INT

`<LLimit>`: Valor limite inferior
Tipo: REAL

`<ULimit>`: Valor limite superior
Tipo: REAL

`<a0> ... <a3>`: Coeficientes da função de polinômio
Tipo: REAL

`PUTFTOCF`: Chamada da função "Gravação contínua por bloco da correção de ferramenta Online"

Parâmetro:

`<função>`: Número da função de polinômio

Tipo: INT

Nota:
Precisa coincidir com a especificação no `FCTDEF`.

`<valor de referência>`: Valor de referência variável, que deve ser derivado da correção (p. ex. o valor real que sofre alterações).

Tipo: VAR REAL

`<parâmetro de ferramenta>`: Número do parâmetro de desgaste (comprimento 1, 2 ou 3), onde o valor de correção deve ser adicionado.

Tipo: INT

`<canal>`: Número do canal, onde a correção de ferramenta Online deve ser ativada.

Tipo: INT

Nota:
Uma especificação somente é necessária se a correção não deve ser ativada no canal ativo.

`<fuso>`: Número do fuso, onde a correção de ferramenta Online deve ser ativada.

Tipo: INT

Nota:
Uma especificação somente é necessária se, ao invés do rebolo ativo, deve ser corrigido um rebolo não ativo no momento da ferramenta empregada.

`PUTFTOC`: Chamada da função "Gravação discreta da correção de ferramenta Online"

Parâmetro:

`<valor de correção>`: Valor de correção que deve ser adicionado no parâmetro de desgaste.

Tipo: REAL

`<parâmetro de ferramenta>`: veja `PUTFTOCF`

`<canal>`: Número do canal, onde a correção de ferramenta Online deve ser ativada.

Tipo: INT

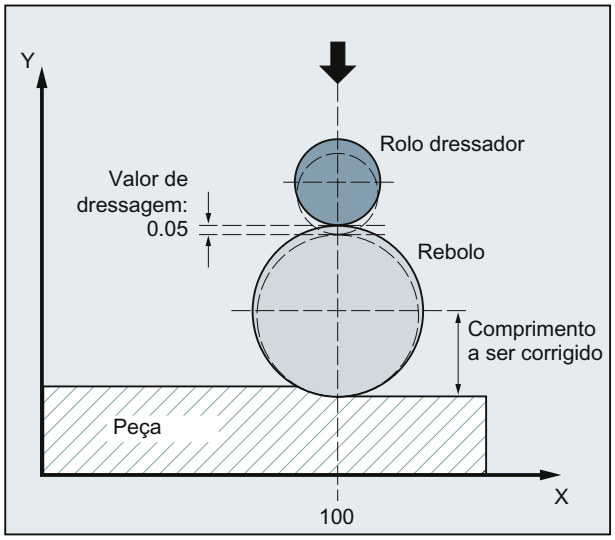
`<fuso>`: veja `PUTFTOCF`

Exemplo

Retificadora plana com:

- Y: Eixo de penetração do rebolo
- V: Eixo de penetração do rolo dressador
- Canal de usinagem: Canal 1 com os eixos X, Y, Z
- Canal de dressagem: Canal 2 com eixo V

Depois de ser iniciado o movimento de retificação, em X100 o rebolo deve ser dressado pelo valor 0,05. O valor de dressagem deve estar ativado na ferramenta para retificação com a "Gravação contínua da correção de ferramenta Online".



Programa de usinagem no canal 1:

Código de programa	Comentário
…	
N110 G1 G18 F10 G90	; Posição inicial.
N120 T1 D1	; Selecionar atual ferramenta.
N130 S100 M3 X100	; Fuso ligado, deslocar até a posição de saída.
N140 INIT(2,"DRESSAGEM","S")	; Seleção do programa de dressagem no canal 2.
N150 START(2)	; Início do programa de dressagem no canal 2.
N160 X200	; Deslocamento até a posição de destino.
N170 FTOCON	; Ativar a correção Online.
N… G1 X100	; Outra usinagem.
N… M30	

Programa de dressagem no canal 2:

```
Código de programa                                  Comentário
…
N40 FCTDEF(1,-1000,1000,-$AA_IW[V],1)               ; Definição de função: Reta com
                                                      inclinação=1.
N50 PUTFTOCF(1,$AA_IW[V],3,1)                       ; Gravação contínua da correção de
                                                      ferramenta Online: derivado do
                                                      movimento do eixo V, o comprimento 3
                                                      do atual rebolo é corrigido no canal
                                                      1.
N60 V-0.05 G1 F0.01 G91                             ; Movimento de penetração para
                                                      dressagem, somente neste bloco em
                                                      que o PUTFTOCF está ativo.
…
N… M30
```

Outras informações

Generalidades sobre a correção de ferramenta Online

Na gravação contínua (para cada ciclo IPO), após a ativação da função de avaliação, qualquer alteração será processada de modo aditivo na memória de desgaste (para evitar saltos de valor nominal).

Em todo caso aplica-se: A correção online da ferramenta pode atuar em cada canal para cada fuso e o comprimento 1, 2 **ou** 3 do parâmetro de desgaste.

A associação dos comprimentos aos eixos geométricos é realizada com base no atual plano de trabalho.

A associação do fuso com a ferramenta é realizada através dos dados de ferramenta com GWPSON ou TMON, enquanto não se tratar do rebolo ativo.

Sempre se corrige o parâmetro de desgaste para o atual lado do rebolo ou o lado esquerdo do rebolo em ferramentas não ativas.

Indicação

Para uma correção idêntica para vários lados do rebolo deve-se providenciar, através de uma diretriz de encadeamento, para que os valores sejam transferidos automaticamente para o segundo lado do rebolo.

Se forem especificadas correções Online para um canal de usinagem, então os valores de desgaste da atual ferramenta não podem ser alterados neste canal a partir de um programa de usinagem ou através da operação.

A correção de ferramenta Online também é considerada na velocidade periférica constante do rebolo (SUG) assim como para a monitoração de ferramentas (TMON).

7.5 Ativação da correção de ferramenta 3D (CUT3DC..., CUT3DF...)

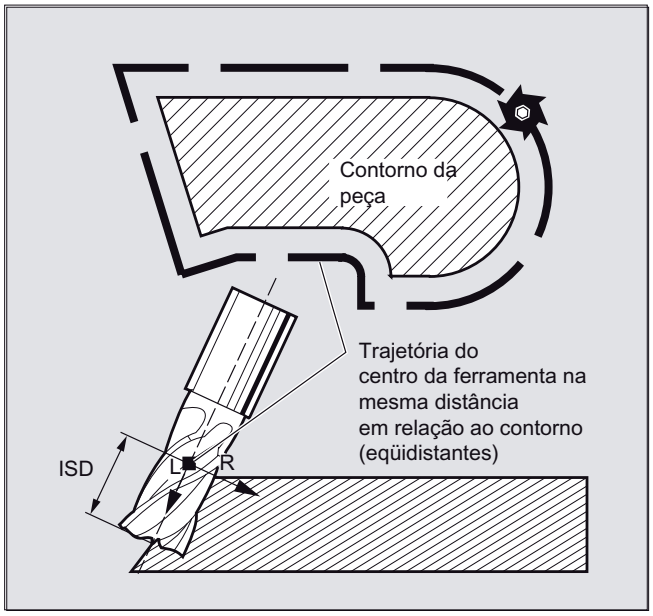
7.5.1 Ativação das correções de ferramentas 3D (CUT3DC, CUT3DF, CUT3DFS, CUT3DFF, ISD)

Função

Na correção do raio da ferramenta para ferramentas cilíndricas é considerada a orientação mutável de ferramenta.

Para seleção da correção do raio da ferramenta 3D são aplicados os mesmos comandos de programação da correção do raio da ferramenta 2D. Com `G41`/`G42` é especificada a correção esquerda/direita no sentido do movimento. O comportamento de aproximação sempre é `NORM`. A correção do raio da ferramenta 3D apenas tem efeito com a transformação de 5 eixos ativada.

A correção do raio da ferramenta 3D também é denominada de correção 5D, pois neste caso temos disponíveis 5 graus livres para a posição da ferramenta no espaço.



Diferença entre as correções de raio da ferramenta 2 1/2D e 3D

Na correção do raio da ferramenta 3D é requisitada a orientação da ferramenta. Na correção do raio da ferramenta 2 1/2D apenas se calcula uma ferramenta com orientação constante.

Sintaxe

```
CUT3DC
CUT3DFS
CUT3DFF
CUT3DF
ISD=<valor>
```

Significado

`CUT3DC`	Ativação da correção do raio da ferramenta 3D para fresamento periférico
`CUT3DFS`	Correção de ferramenta 3D para fresamento de topo com orientação constante. A orientação da ferramenta é definida pelo `G17` - `G19` e não influenciada por Frames.
`CUT3DFF`	Correção de ferramenta 3D para fresamento de topo com orientação constante. A orientação da ferramenta é definida pelo `G17` - `G19` e, se necessário, através de um sentido girado por um Frame.
`CUT3DF`	Correção de ferramenta 3D para fresamento de topo com alteração de orientação (apenas na transformação de 5 eixos ativa).
`G40 X... Y... Z...`	Para desativar: Bloco linear `G0`/`G1` com eixos geométricos
`ISD`	Profundidade de imersão

Indicação

Os comandos estão ativos de forma modal e estão no mesmo grupo que o `CUT2D` e `CUT2DF`. A desseleção apenas é realizada com o próximo movimento do atual plano. Isto sempre se aplica ao `G40` e não depende do comando `CUT`.

São permitidos blocos intermediários com a correção ativa do raio da ferramenta 3D. São aplicadas as definições da correção do raio da ferramenta 2 1/2D.

Condições gerais

- **G450/G451 e DISC**

  Nos cantos externos sempre é inserido um bloco circular. `G450/G451` são irrelevantes. O comando `DISC` não é avaliado.

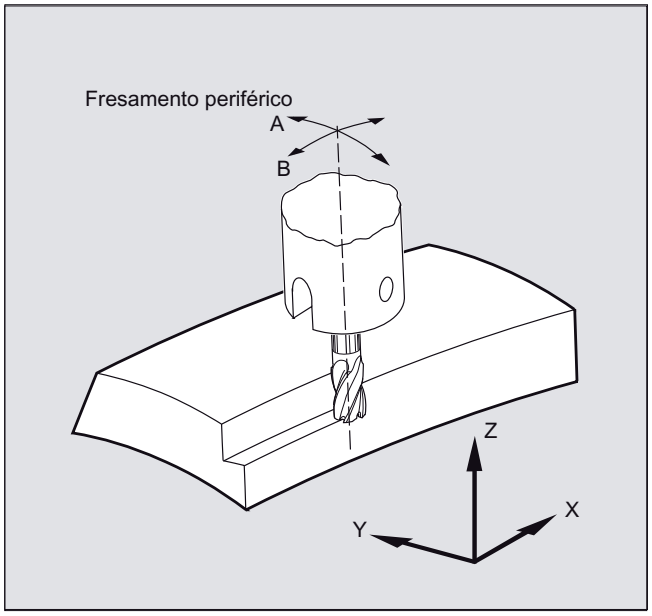
Exemplo

```
Código de programa                Comentário
N10 A0 B0 X0 Y0 Z0 F5000
N20 T1 D1                         ; Chamada da ferramenta, chamar valores de correção
                                    da ferramenta.
N30 TRAORI(1)                     ; Seleção da transformação
N40 CUT3DC                        ; Seleção da correção do raio da ferramenta 3D
N50 G42 X10 Y10                   ; Seleção da correção do raio da ferramenta
N60 X60
N70 …
```

7.5.2 Correções de ferramenta 3D: Fresamento periférico, fresamento de topo

Fresamento periférico

A variante de fresamento periférico utilizada neste caso é realizada com a especificação de um percurso (linha guia) e da orientação correspondente. Neste tipo de usinagem o tipo de ferramenta no percurso é irrelevante. O único fator decisivo é o raio no ponto de ataque da ferramenta.

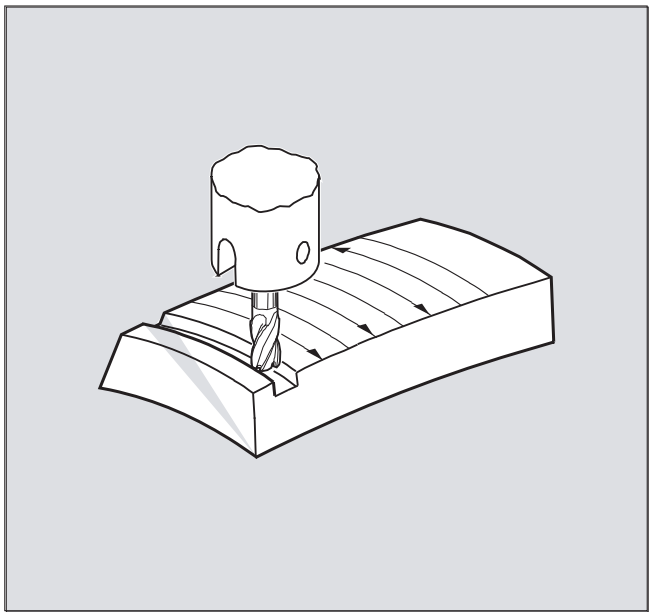


Indicação

A função de correção do raio da ferramenta 3D limita-se a ferramentas cilíndricas.

Fresamento de topo

Para este tipo de fresamento 3D precisamos da descrição das trajetórias 3D sobre a superfície da peça de trabalho. Normalmente no CAM os cálculos são executados considerando-se a forma e dimensões da ferramenta. O pós-processador, além dos blocos NC, grava no programa de peças as orientações da ferramenta (com transformação de 5 eixos ativada) e o código G da correção de ferramenta 3D desejada. Com isso o operador da máquina tem a opção de empregar ferramentas muito menores do que as empregadas no cálculo dos percursos NC.



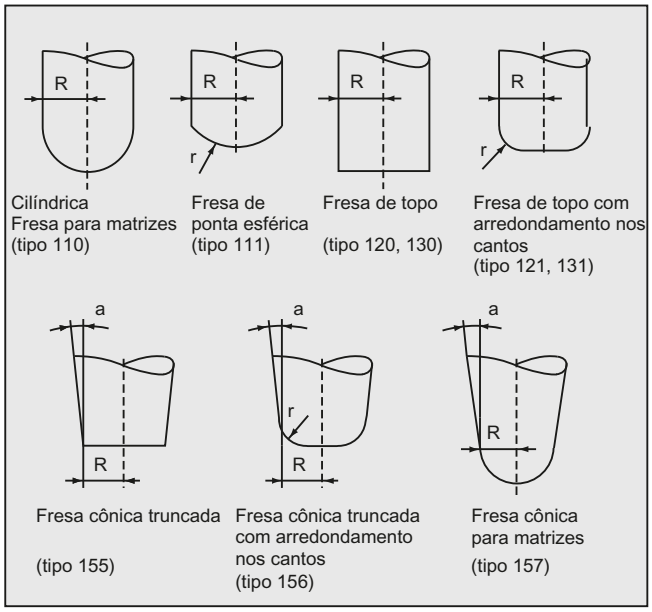
Exemplo:

Os blocos NC foram calculados com uma fresa de 10 mm. Aqui também se poderia usinar uma fresa de diâmetro 9,9 mm, sendo que neste caso se deve considerar uma possível alteração no perfil de rugosidade.

7.5.3 Correções de ferramenta 3D: Formas e dados de ferramenta para fresamento de topo

Formas de fresas, dados de ferramenta

Na seguinte tabela estão agrupadas as possíveis formas de ferramenta e os valores limite dos dados de ferramenta para o fresamento de topo. A forma da haste da ferramenta não é considerada. Os tipos de ferramenta 120 e 156 são idênticas em efeito.



Se no programa NC for especificado um número de tipo diferente dos indicados na figura, o sistema aplica automaticamente o tipo de ferramenta 110 (fresa cilíndrica para matrizes). É emitido um alarme se os valores limite dos dados da ferramenta forem violados.

Tipo de fresa	Tipo nº	R	r	a
Fresa cilíndrica para matrizes	110	> 0	-	-
Fresa de ponta esférica	111	> 0	>R	-
Fresa de topo, fresa angular	120, 130	> 0	-	-
Fresa de topo, fresa angular com cantos arredondados	121, 131	>r	> 0	-
Fresa cônica truncada	155	> 0	-	> 0
Fresa cônica truncada com arredondamento nos cantos	156	> 0	> 0	> 0
Fresa cônica para matrizes	157	> 0	-	> 0

R = raio da haste (raio da ferramenta)

r = raio do canto

a = ângulo entre o eixo longitudinal da ferramenta e o canto superior da superfície do toro

- = não é avaliado

Dados de ferramenta	Parâmetro da ferramenta	
Dimensões da ferramenta	Geometria	Desgaste
R	$TC_DP6	$TC_DP15
r	$TC_DP7	$TC_DP16
a	$TC_DP11	$TC_DP20

Correção do comprimento da ferramenta

Com o ponto de referência da correção do comprimento se aplica a ponta da ferramenta (ponto de corte do eixo longitudinal/superfície).

Correção da ferramenta 3D, troca de ferramentas

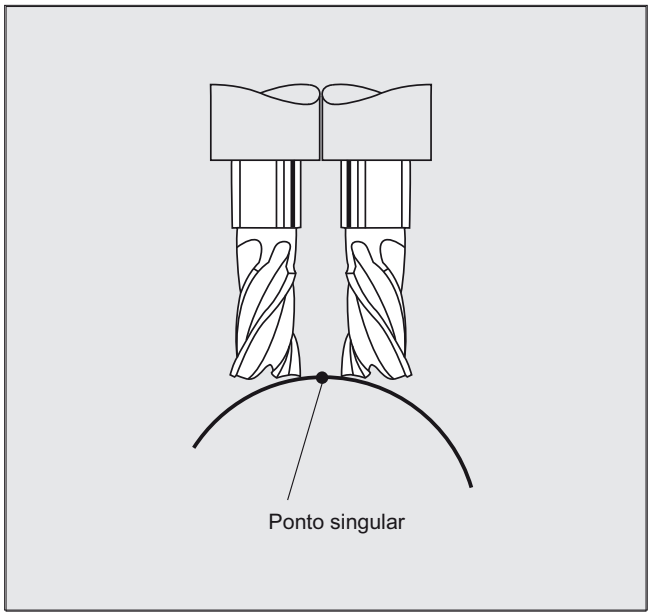
Uma ferramenta nova com dimensões alteradas (R, r, a) ou outra forma somente pode ser especificada com a programação do `G41` ou `G42` (transição do `G40` para `G41` ou `G42`, nova programação do `G41` ou `G42`). Todos demais dados de ferramenta, como comprimento de ferramenta, não são considerados por esta regra, para que tais ferramentas também possam ser trocadas sem um novo `G41` ou `G42`.

7.5.4 Correções de ferramenta 3D: Correção na trajetória, curvatura de trajetória, profundidade de imersão (CUT3DC, ISD)

Função

Correção na trajetória

No fresamento de topo se deve considerar o caso em que o ponto de contato salta sobre a superfície da ferramenta. Como neste exemplo de usinagem de uma superfície convexa com ferramenta em posição vertical. A aplicação indicada na figura pode ser considerada como caso limite.



Este caso limite é monitorado pelo comando, onde são detectadas alterações bruscas do ponto de corte com base nas posições angulares entre a ferramenta e os vetores normais da superfície. Nestas posições o comando insere blocos lineares de modo que o movimento possa ser executado.

Para calcular os blocos lineares estão armazenadas faixas angulares permitidas em dados da máquina para o ângulo lateral. Se os valores de limite definidos em dados da máquina forem ultrapassados nas faixas angulares, então o sistema emitirá um alarme.

Curvatura de trajetória

A curvatura de trajetória não é monitorada. Também aqui se recomenda apenas empregar ferramentas com as quais se pode trabalhar sem danificar o contorno.

Profundidade de imersão (ISD)

A profundidade de imersão ISD somente é avaliada na correção do raio da ferramenta 3D.

Com o comando de programa `ISD` (Insertion depth) se programa a profundidade de imersão da ferramenta no fresamento periférico. Com isso é possível alterar a posição do ponto de corte sobre a superfície envolvente da ferramenta.

Sintaxe

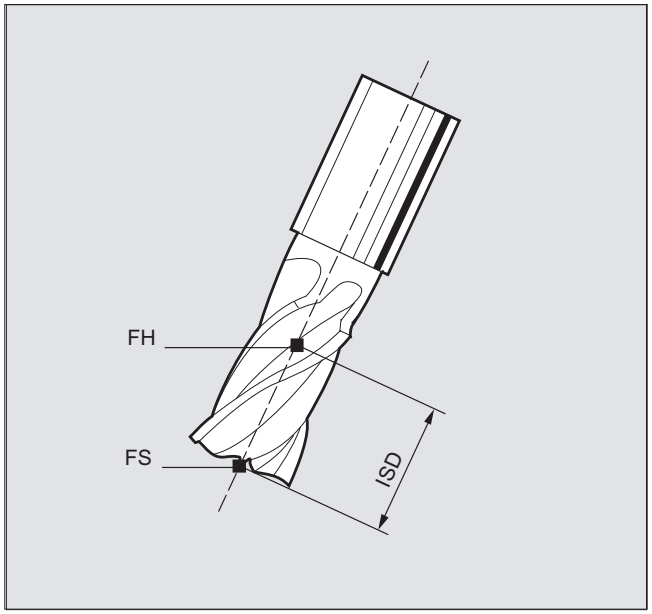
Correções de ferramenta 3D no fresamento periférico
```
CUT3DC
ISD=<valor>
```

Significado

`CUT3DC`	Ativar a correção de ferramenta 3D para fresamento periférico, p. ex. para fresamento de bolsões com paredes laterais inclinadas.
`ISD`	Com o comando `ISD` se especifica a distância (`<valor>`) entre a ponta da fresa (FS) e o ponto auxiliar da fresa (FH).

Ponto auxiliar da fresa

O ponto auxiliar da fresa (FH) é formado pela projeção do ponto de usinagem programado no eixo da ferramenta.



Outras informações

Fresamento de bolsões com paredes laterais inclinadas para fresamento periférico com CUT3DC

Nesta correção do raio da ferramenta 3D se compensa o desvio do raio da ferramenta ao passo que o avanço é realizado no sentido da normal da superfície com a superfície usinada. Aqui o plano onde está o lado frontal da fresa permanece inalterado se a profundidade de imersão `ISD` for a mesma. Por exemplo, se comparada à uma ferramenta normalizada, uma fresa de raio menor não alcançaria a base do bolsão, este que também forma a superfície de limitação. Para uma penetração automática da ferramenta a superfície de limitação deve ser conhecida do comando, veja o capítulo "Fresamento periférico 3D com superfícies de limitação".

Para mais informações sobre a monitoração de colisões, veja:
Literatura:
Manual de programação Fundamentos; Capítulo "Correções de ferramenta".

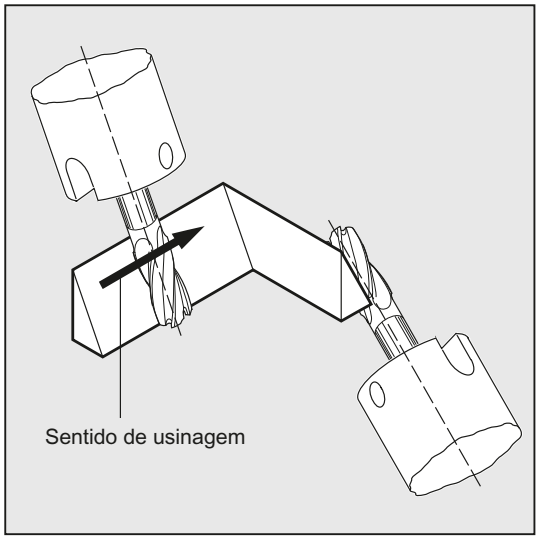
7.5.5 Correções de ferramenta 3D: Cantos internos/externos e procedimento de ponto de intersecção (G450/G451)

Função

Cantos internos/externos

Os cantos externos e internos são tratados separadamente. A denominação canto interno ou canto externo depende da orientação da ferramenta.

Nas alterações de orientação em um canto pode ocorrer que o tipo de canto mude durante a usinagem. Se isto ocorrer, então a usinagem será cancelada com uma mensagem de erro.



Sintaxe

```
G450
G451
```

Significado

G450	Círculo de transição (a ferramenta percorre os cantos da peça em uma trajetória circular)
G451	Ponto de intersecção das eqüidistantes (a ferramenta usina para retirada do canto da peça)

Outras informações

Procedimento do ponto de intersecção para correção 3D

Agora para o fresamento periférico 3D se avalia o código G450/G451 nos cantos externos, ou seja, se pode aproximar o ponto de intersecção das curvas de deslocamento. Até o SW 4 sempre foi inserido um círculo nos cantos externos. O procedimento do ponto de intersecção disponível é principalmente vantajoso nos programas 3D gerados em CAD. Estes muitas vezes são constituídos de curtos blocos lineares (para aproximação de curvas lisas), nos quais as transições são quase tangenciais entre os blocos adjacentes.

Na correção do raio da ferramenta no lado externo do contorno até então sempre foram inseridos círculos para contornar os cantos externos. Visto que estes blocos se tornam muito curtos quando se aproximam das transições tangenciais, resultam interrupções inconvenientes de velocidade.

Nestes casos as duas curvas envolvidas são prolongadas de modo similar à correção do raio 2 ½ D, se aproxima a intersecção de ambas curvas prolongadas.

A intersecção é definida quando as curvas de deslocamento de ambos blocos envolvidos são prolongadas e sua intersecção definida no canto perpendicular à orientação da ferramenta no plano. Se não houver uma intersecção deste tipo, o canto será tratado como antes, isto é, se insere um círculo.

Para mais informações sobre procedimentos de ponto de intersecção, veja:
Literatura:
Manual de funções especiais; Correção do raio da ferramenta 3D (W5)

7.5.6 Correções de ferramenta 3D: Fresamento periférico 3D com superfícies de limitação

Adaptações do fresamento periférico 3D às condições dos programas CAD

Os programas NC gerados por sistemas CAD normalmente aproximam a trajetória da linha de centro de uma ferramenta normalizada com um grande número de blocos lineares curtos. Para que os blocos gerados dessa forma possam reproduzir vários contornos parciais o mais próximo do contorno original, torna-se necessário realizar algumas adaptações no programa de peça.

As informações importantes que são necessárias para uma correção ideal, mas que não estão mais disponíveis no programa de peça, devem ser substituídas por medidas adequadas. A seguir são representados os métodos típicos para compensar transições críticas diretamente no programa de peça ou na determinação do contorno real (p. ex. através da penetração da ferramenta).

Aplicações

Além dos casos típicos de aplicação, onde uma ferramenta real descreve a trajetória da linha de centro ao invés da ferramenta normalizada, também são tratadas as ferramentas cilíndricas com correção de ferramenta 3D. Aqui a trajetória programada toma o contorno na superfície de usinagem como referência. A superfície de limitação neste caso é independente da ferramenta. Como na correção convencional do raio da ferramenta, é utilizado o raio total para o cálculo do deslocamento perpendicular com a superfície de limitação.

7.5.7 Correções de ferramenta 3D: Consideração de uma superfície de limitação (CUT3DCC, CUT3DCCD)

Função

Fresamento periférico 3D com ferramentas reais

No fresamento periférico 3D com alteração contínua ou constante da orientação da ferramenta, muitas vezes se programa o percurso do centro da ferramenta para uma ferramenta normalizada definida. Visto que na prática estas ferramentas normalizadas correspondentes muitas vezes não estão à disposição, não se pode empregar uma ferramenta muito diferente da ferramenta normalizada.

Com `CUT3DCCD` se considera uma superfície de limitação para uma ferramenta diferencial, que descreveria a ferramenta normalizada programada. O programa NC descreve a trajetória do centro da ferramenta normalizada.

Com `CUT3DCC` se considera uma superfície de limitação com o uso de ferramentas cilíndricas, que teria alcançado a ferramenta normalizada programada. O programa NC descreve o contorno na superfície de usinagem.

Sintaxe

```
CUT3DCCD
CUT3DCC
```

Significado

`CUT3DCCD`	Ativação da correção de ferramenta 3D para fresamento periférico com superfícies de limitação com ferramenta diferencial na trajetória do centro da ferramenta: Penetração até a superfície de limitação.
`CUT3DCC`	Ativação da correção de ferramenta 3D para fresamento periférico com superfícies de limitação com correção de raio 3D: Contorno na superfície de usinagem

Indicação

Correção do raio de ferramenta com G41, G42

Para correção do raio de ferramenta com `G41`, `G42` com `CUT3DCCD` ou `CUT3DCC` ativo deve estar disponível o opcional "Transformação de orientação".

Ferramenta normalizada com arredondamento de cantos

O arredondamento de cantos da ferramenta normalizada é descrito pelo parâmetro de ferramenta `$TC_DP7`. A partir do parâmetro de ferramenta `$TC_DP16` resulta o desvio do arredondamento de canto da ferramenta real comparado com a ferramenta normalizada.

Exemplo

As dimensões de ferramenta de uma fresa toroidal com raio reduzido se comparada com uma ferramenta normalizada.

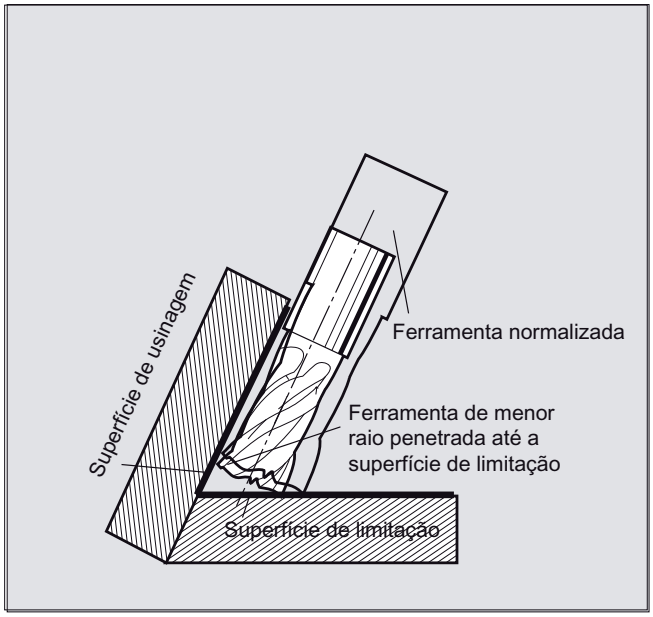
Tipo de ferramenta	R = raio da haste	r = raio de canto
Ferramenta normalizada com arredondamento de cantos	R = $TC_DP6	r = $TC_DP7
Ferramenta real com arredondamento de cantos: Tipos de ferramenta 121 e 131, fresa toroidal (fresa de topo)	R' = $TC_DP6 + $TC_DP15 + OFFN	r' = $TC_DP7 + $TC_DP16
Neste exemplo tanto o $TC_DP15 + OFFN como o $TC_DP16 são negativos. O tipo de ferramenta ($TC_DP1) é avaliado.		
Permitidos estão apenas os tipos de fresa com haste cilíndrica (fresas cilíndricas ou fresas de topo) assim como fresas toroidais (tipo 121 e 131) no caso limite da fresa cilíndrica para matrizes (tipo 110).	Nestes tipos permitidos de fresa o raio do canto r é igual ao raio da haste R. Todos outros tipos de ferramenta permitidos são interpretados como fresas cilíndricas e uma eventual dimensão especificada para arredondamento de canto não será avaliada.	

Tipo de ferramenta	R = raio da haste	r = raio de canto
São permitidos todos tipos de ferramentas de número 1 – 399 com exceção dos números 111 e 155 até 157.		

Outras informações

Trajetória do centro da ferramenta com penetração até a superfície de limitação CUT3DCCD

Se for utilizada uma ferramenta com um raio menor em relação à ferramenta normalizada correspondente, então uma fresa penetrada no sentido longitudinal será conduzida até que atinja novamente a base do bolsão. Com isso o canto formado pela superfície de usinagem e da superfície de limitação será desbastado até onde a ferramenta permitir. Neste caso trata-se de um modo de usinagem misto de fresamento periférico e fresamento de topo. Semelhante à uma ferramenta com raio reduzido, para uma ferramenta com raio aumentado, a penetração é realizada no sentido contrário.


Superfície de usinagem
Ferramenta normalizada
Ferramenta de menor raio penetrada até a superfície de limitação
Superfície de limitação

Em comparação com as outras correções da ferramenta do grupo 22 de códigos G, um parâmetro de ferramenta `$TC_DP6` especificado para `CUT3DCCD` não é relevante para o raio da ferramenta e não tem influência sobre a correção resultante.

O deslocamento de correção resulta da soma do:

- Valor de desgaste do raio da ferramenta (parâmetro de ferramenta `$TC_DP15`)
- e um Offset de ferramenta `OFFN` para o cálculo do Offset perpendicular à superfície de limitação.

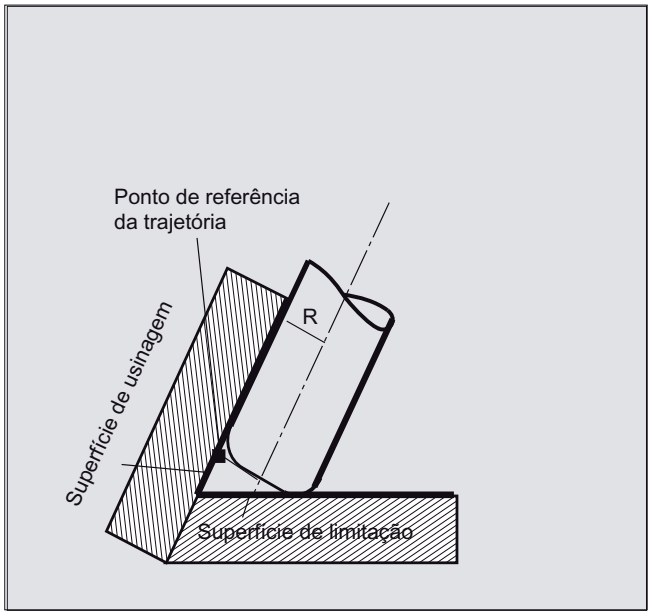
A partir do programa de peças criado não é possível saber se a superfície de usinagem está à esquerda ou à direita do percurso. Por isso que se parte do princípio de um raio positivo e um valor negativo de desgaste da ferramenta original. Um valor negativo de desgaste sempre descreve uma ferramenta com diâmetro reduzido.

Utilização de ferramentas cilíndricas

Na utilização de ferramentas cilíndricas uma penetração apenas será necessária se a superfície de usinagem e a superfície de limitação formarem um ângulo agudo (menor que 90 graus). Se forem utilizadas fresas toroidais (cilindro com arredondamento de cantos), então a penetração no sentido longitudinal da ferramenta será necessária tanto em ângulos agudos como obtusos.

Correção de raio 3D com CUT3DCC, contorno na superfície de usinagem

Se `CUT3DCC` estiver ativo com uma fresa toroidal, então o percurso programado toma como referência uma fresa cilíndrica fictícia de mesmo diâmetro. O ponto de referência do percurso resultante é representado na figura a seguir com a utilização de uma fresa toroidal.



É permitido que o ângulo entre as superfícies de usinagem e de limitação também pode passar de um ângulo agudo para um ângulo obtuso ou vice-versa dentro de um bloco.

Ao contrário da ferramenta normalizada, a ferramenta real utilizada pode ser tanto maior do que menor. Neste caso o raio de canto resultante não pode se tornar negativo e o sinal do raio de ferramenta resultante deve ser preservado.

No `CUT3DCC` o programa NC de peça tem sua referência no contorno da superfície de usinagem. Aqui, como na correção convencional do raio da ferramenta, se chama o raio total formado pela soma do:

- Raio de ferramenta (parâmetro de ferramenta `$TC_DP6`)
- Valor de desgaste (parâmetro de ferramenta `$TC_DP15`)
- e um Offset de ferramenta `OFFN` para o cálculo do Offset perpendicular à superfície de limitação.

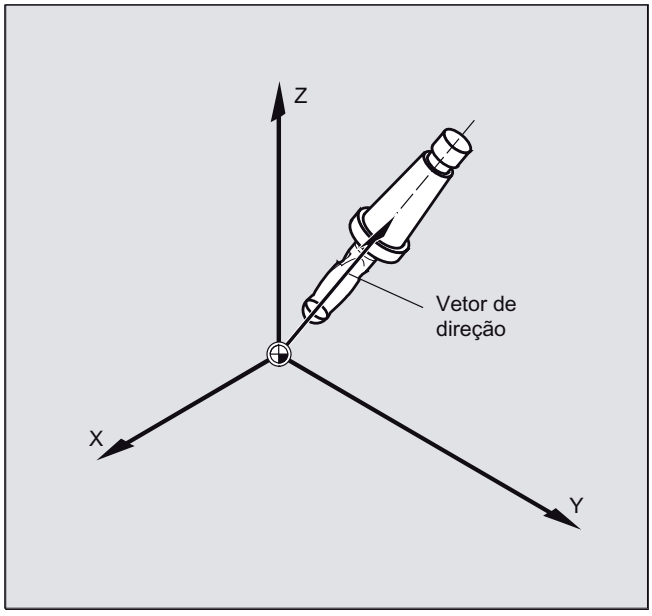
A posição da superfície de limitação é definida pela diferença de ambos valores:

- Dimensões da ferramenta normalizada
- Raio de ferramenta (parâmetro de ferramenta `$TC_DP6`)

7.6 Orientação da ferramenta (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST)

Função

Como orientação da ferramenta entendemos o alinhamento geométrico da ferramenta no espaço. Em uma máquina de usinagem de 5 eixos a orientação da ferramenta pode ser ajustada através de comandos de programa.



Os movimentos de suavização ativados com `OSD` e `OST` são formados de modo diferente dependendo do tipo de interpolação da orientação da ferramenta.

Com a interpolação de vetor ativa também se interpola o decurso de orientação suavizado através da interpolação de vetor. Em contrapartida, na interpolação de eixos rotativos ativa, a orientação é suavizada através de movimentos de eixo rotativo.

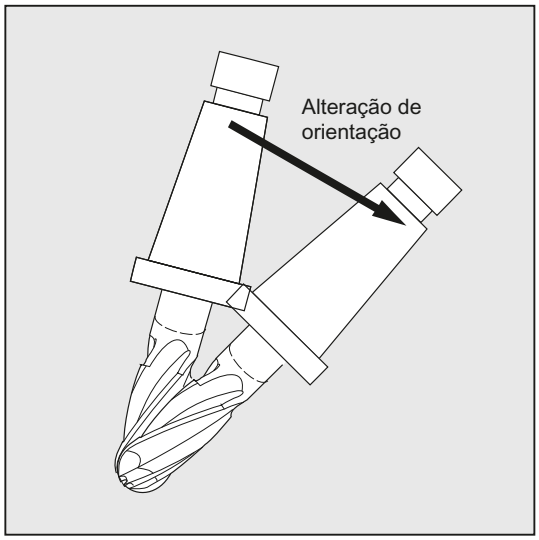
Programação

Programação da alteração de orientação:

Uma alteração da orientação da ferramenta pode ser programada através de:

- programação direta dos eixos rotativos `A`, `B` e `C` (interpolação de eixos rotativos)
- Ângulo euleriano ou ângulo RPY
- Vetor de direção (interpolação de vetor através da especificação do `A3` ou `B3` ou `C3`)
- `LEAD/TILT` (fresamento de topo)

O sistema de coordenadas de referência é o sistema de coordenadas da máquina (`ORIMKS`) ou o atual sistema de coordenadas da peça (`ORIWKS`).



Programação da orientação da ferramenta:

Comando	Significado
ORIC:	Orientação e movimento de percurso paralelo
ORID:	Orientação e movimento de percurso sucessivamente
OSOF:	sem suavização da orientação
OSC:	Orientação constante
OSS:	Suavização da orientação apenas no começo do bloco
OSSE:	Suavização da orientação no começo e no fim do bloco
ORIS:	Velocidade da alteração da orientação com suavização de orientação ativada em graus por mm (vale para OSS e OSSE)
OSD:	Suavização da orientação através da especificação da extensão de suavização com o dado de ajuste: SD42674 $SC_ORI_SMOOTH_DIST
OST:	Suavização da orientação através da especificação da tolerância angular em graus na interpolação de vetor com o dado de ajuste: SD42676 $SC_ORI_SMOOTH_TOL Na interpolação de eixos rotativos se assume a tolerância especificada como o desvio máximo dos eixos de orientação.

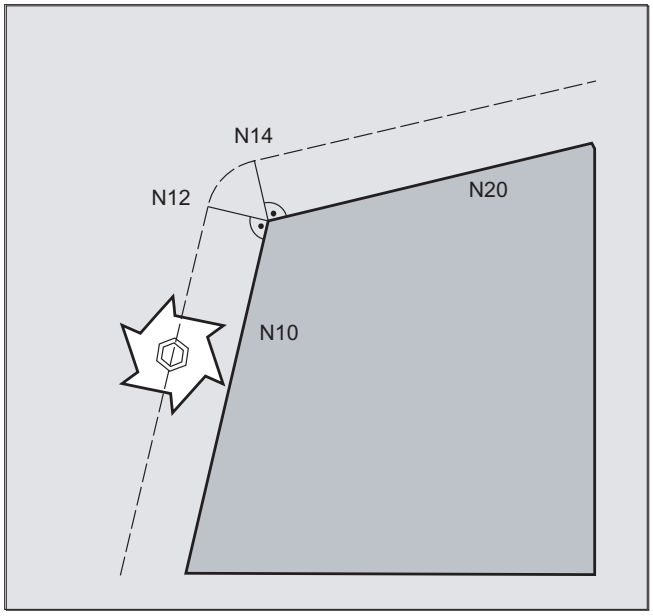
Indicação

Todos comandos para suavização da orientação de ferramenta (OSOF, OSC, OSS, OSSE, OSD e OST) estão agrupados no grupo de funções G 34. Elas estão ativas modalmente, ou seja, sempre apenas um destes comandos poderá atuar.

Exemplos

Exemplo 1: ORIC

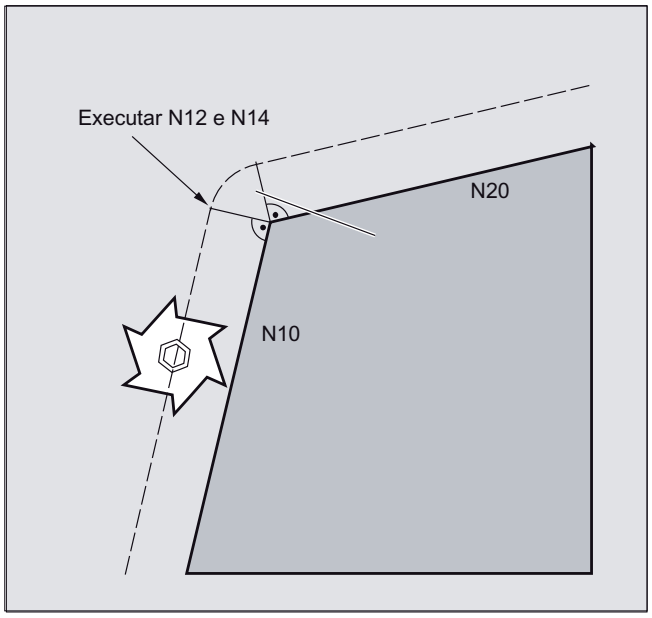
Se entre os blocos de deslocamento `N10` e`N20`foram programados dois ou mais blocos com alterações de orientação (p. ex. `A2=... B2=... C2=...`) e o `ORIC` estiver ativo, então o bloco circular inserido será divido nestes blocos intermediários de acordo com o valor das alterações angulares.



Código de programa	**Comentário**
ORIC	
N8 A2=… B2=… C2=…	
N10 X… Y… Z…	
N12 C2=… B2=…	; O bloco circular inserido no canto externo
N14 C2=… B2=…	distribui-se entre o N12 e o N14, conforme a alteração da orientação. Neste caso o movimento circular e a alteração da orientação são executados paralelamente.
N20 X =…Y=… Z=… G1 F200	

Exemplo 2: ORID

Se o `ORID` estiver ativo, então serão executados todos blocos entre os dois blocos de deslocamento no fim do primeiro bloco de deslocamento. O bloco circular com orientação constante é executado imediatamente antes do segundo bloco de deslocamento.



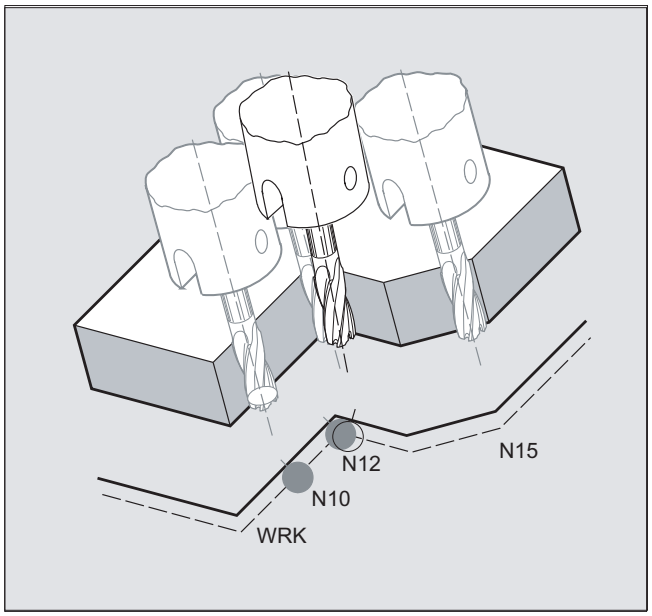
Código de programa	Comentário
`ORID`	
`N8 A2=… B2=… C2=…`	
`N10 X… Y… Z…`	
`N12 A2=… B2=… C2=…`	`; O bloco N12 e N14 é executado no fim do N10. Em seguida é executado o bloco circular com a atual orientação.`
`N14 M20`	`; Funções auxiliares, etc.`
`N20 X… Y… Z…`	

Indicação

Para o tipo da alteração da orientação em um canto externo será determinante o comando de programa que estiver ativo no primeiro bloco de deslocamento de um canto externo.

Sem alteração da orientação: Se a orientação no limite de bloco não for alterada, então a secção transversal da ferramenta será um círculo que tem contato em ambos contornos.

Exemplo 3: Alteração da orientação em um canto interno



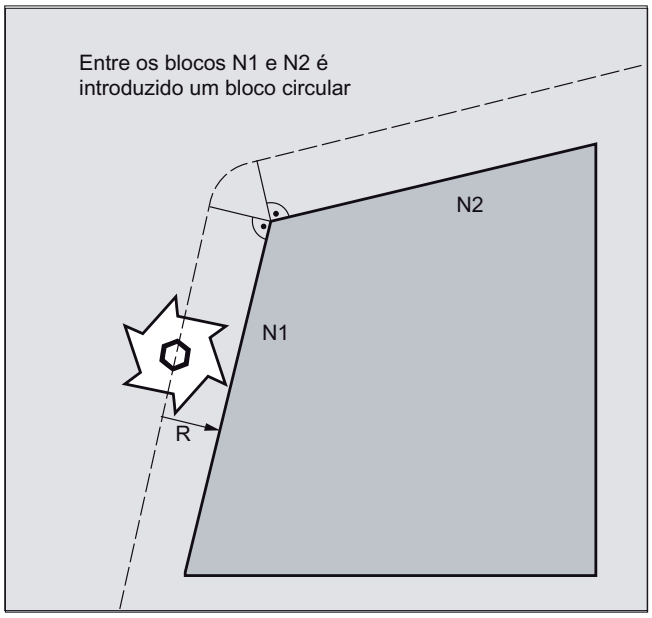
Código de programa
ORIC
N10 X …Y… Z… G1 F500
N12 X …Y… Z… A2=… B2=… C2=…
N15 X …Y… Z… A2=… B2=… C2=…

Outras informações

Comportamento em cantos externos

Em um canto externo sempre é inserido um bloco circular com o raio da fresa.

Com os comandos de programa `ORIC` ou `ORID` se pode definir se as alterações de orientação que foram programadas entre os blocos `N1` e `N2` serão executadas antes do início do bloco circular inserido, ou simultaneamente com este.



Se uma alteração de orientação for necessária em cantos externos, então ela pode ser realizada paralelamente à interpolação ou separadamente com o movimento de percurso.

Com ORID são executados primeiramente os blocos inseridos sem movimento de percurso. O bloco circular é inserido imediatamente antes do segundo dos dois blocos que formam o canto.

Se em um canto externo foram inseridos vários blocos de orientação e selecionado o ORIC, então o movimento circular será distribuído sobre os vários blocos de acordo com os valores das alterações de orientação dos diversos blocos inseridos.

Suavização da orientação com OSD ou OST

Na suavização com G642 o desvio máximo para eixos de contorno e eixos de orientação não pode ser muito diferente. A menor tolerância dos dois determina a forma do movimento de suavização ou tolerância angular, para suavizar o decurso de orientação relativamente forte, sem precisar de maiores desvios de contorno.

Através da ativação do OSD ou do OST é possível, com uma extensão de suavização especificada ou tolerância angular, suavizar com facilidade desvios muito pequenos do decurso de orientação sem desvios de contorno muito grandes.

Indicação

Diferente da suavização do contorno (e do decurso de orientação) com G642, na suavização da orientação com OSD ou OST não se forma nenhum bloco próprio, mas se insere o movimento da suavização diretamente nos blocos originais programados.

Com OSD ou OST não pode ser suavizada nenhuma transição de blocos onde existe uma mudança de tipos de interpolação para orientação da ferramenta (vetor → eixo rotativo, eixo rotativo → vetor). Estas transições de blocos podem, eventualmente, ser suavizados com as funções convencionais de suavização G641, G642 e G643.

7.7 Seleção livre de número D, número de gume

7.7.1 Atribuição livre de números D, número de corte (endereço CE)

Número D

Os números D podem ser utilizados como números de correção. Além disso se pode endereçar o número do corte através do endereço CE. Através da variável de sistema $TC_DPCE se pode descrever o número de corte.

Pré-ajuste: Número de correção == número de cortes

Através dos dados de máquina se pode definir o número máximo de números D (números de corte) e o número máximo de cortes por ferramenta (→ Fabricante da máquina). Os comandos a seguir apenas têm relevância se o número máximo de cortes (MD18105) for definido maior do que o número de cortes por ferramenta (MD18106). Observe as informações do fabricante da máquina.

Indicação

Além da atribuição relativa de números D, os números D também podem ser atribuídos como números D "planos" ou "absolutos" (1-32000) sem referência a um número T (dentro da função "Estrutura plana de números D").

Literatura

Manual de funções básicas; Correção de ferramenta (W1)

7.7.2 Atribuição livre de números D: Verificar números D (CHKDNO)

Função

Com o comando `CKKDNO` verificamos se os números D presentes foram especificados de forma única. Os números D de todas ferramentas definidas dentro de uma unidade TO podem aparecer apenas uma vez. As ferramentas de reposição não são consideradas neste caso.

Sintaxe

```
state=CHKDNO(Tno1,Tno2,Dno)
```

Significado

`state`	=TRUE:	Os números D foram especificados exclusivamente para a faixa verificada.
	=FALSE:	Ocorreu uma colisão de números D ou a parametrização é inválida. Através de Tno1, Tno2 e Dno são transmitidos os parâmetros que geram a colisão. Estes dados podem ser avaliados no programa de peça.
`CHKDNO(Tno1,Tno2)`	São verificados todos os números D das ferramentas mencionadas.	
`CHKDNO(Tno1)`	São verificados todos números D de Tno1 contra todas outras ferramentas.	
`CHKDNO`	São verificados todos números D de todas ferramentas contra todas outras ferramentas.	

7.7.3 Atribuição livre de números D: Renomear números D (GETDNO, SETDNO)

Função

Os números D devem ser especificados como únicos. Dois cortes diferentes de uma ferramenta não podem ter o mesmo número D.

GETDNO

Este comando retorna o número D de um determinado corte (ce) de uma ferramenta de número T (t). Se não existir nenhum número D para os parâmetros especificados, se define d=0. Se o número D for inválido, será retornado um valor acima de 32000.

SETDNO

Com este comando atribuímos o valor d do número D de um corte ce da ferramenta t. Através do state é retornado o resultado desta instrução (TRUE ou FALSE). Se não houver nenhum bloco de dados para os parâmetros especificados, será retornado FALSE. Os erros de sintaxe geram um alarme. O número D não pode ser passado explicitamente para 0.

Sintaxe

```
d = GETDNO (t,ce)
state = SETDNO (t,ce,d)
```

Significado

`d`	Número D do corte da ferramenta
`t`	Número T da ferramenta
`ce`	Número do corte (número CE) da ferramenta
`state`	Indica se o comando poderia ser executado sem erros (TRUE ou FALSE).

Exemplo de renomeação de um número D

Programação	**Comentário**
`$TC_DP2[1,2]=120`	`;`
`$TC_DP3[1,2] = 5.5`	`;`
`$TC_DPCE[1,2] = 3`	`;  Número de corte CE`
`...`	`;`
`N10 def int DNoAnt, DNoNov = 17`	`;`
`N20 NoDAnt = GETDNO(1,3)`	`;`
`N30 SETDNO(1,3,NoDNov)`	`;`

Com isso o novo valor D 17 é atribuído para o corte CE=3. Agora os dados deste corte são endereçados através do número D 17; tanto através de variáveis de sistema como na programação com o endereço NC.

7.7.4 Atribuição livre de números D: Determinar o número T para o número D especificado (GETACTTD)

Função

Com o comando `GETACTTD` determinamos o número T pertencente a um número F absoluto. Não é realizada nenhuma verificação quanto à exclusividade. Se existirem vários números D iguais dentro de uma unidade TO, será retornado o número T da primeira ferramenta encontrada. Com a utilização de números D "planos" não existe relevância no uso do comando, pois aqui sempre é retornado o valor 1 (nenhum número T no banco de dados).

Sintaxe

```
status=GETACTTD(Tno,Dno)
```

Significado

`Dno`	Número D que deve ser procurado para o número T.	
`Tno`	Número T encontrado	
`estado`	Valor:	Significado:
	0	O número T foi encontrado. O Tno contém o valor do número T.
	-1	Para o número D indicado não existe nenhum número T; Tno=0.
	-2	O número D não é absoluto. O Tno contém o valor da primeira ferramenta encontrada que contém o número D com o valor Dno.
	-5	A função não pôde ser executada por outro motivo.

7.7.5 Atribuição livre de números D: Invalidar números D (DZERO)

Função

O comando DZERO serve para dar suporte durante a mudança de ferramentas. Os blocos de dados de correção marcados não são mais verificados pelo comando CHKDNO. Para acessá-los novamente, o número D deve ser ajustado novamente com SETDNO.

Sintaxe

```
DZERO
```

Significado

DZERO	Marca todos números D da unidade TO como inválidos.

7.8 Cinemática do porta-ferramenta

Requisitos

Um porta-ferramenta somente pode orientar uma ferramenta sem todas possíveis direções no espaço se

- dois eixos de rotação $V1$ e $V2$ estiverem disponíveis.
- os eixos de rotação estão perpendiculares entre si.
- o eixo longitudinal da ferramenta está perpendicular em relação ao segundo eixo de rotação $V2$.

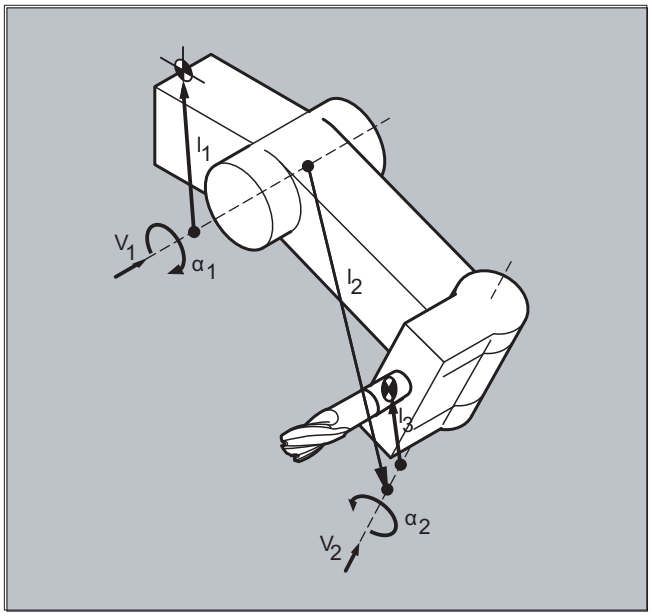
Em máquinas onde todas possíveis orientações devem ser ajustáveis, também se aplica o seguinte:

- a orientação da ferramenta deve estar perpendicular em relação ao primeiro eixo de rotação $V1$.

Função

A cinemática da ferramenta com no máximo dois eixos de rotação v_1 ou v_2 é descrita através das 17 variáveis de sistema `$TC_CARR1[m]` até `$TC_CARR17[m]`. A descrição do porta-ferramenta é composta por:

- a distância vetorial do primeiro eixo de rotação até o ponto de referência do porta-ferramenta $I1$, a distância vetorial do primeiro e segundo eixo de rotação $I2$, a distância vetorial do segundo eixo de rotação até o ponto de referência da ferramenta $I3$.
- os vetores de direção dos dois eixos de rotação $V1$, $V2$.
- dos ângulos de rotação $\alpha1$, $\alpha2$em torno de ambos eixos. Os ângulos de rotação são contados positivos no sentido de visualização dos vetores dos eixos rotativos em sentido horário.



Para máquinas com **cinemática resolvida** (se gira tanto a ferramenta como a peça) foram ampliadas as variáveis de sistema e os registros

- `$TC_CARR18[m]` até `$TC_CARR23[m]` .

Parâmetros

Função das variáveis de sistema para porta-ferramentas orientáveis			
Denominação	**Componente x**	**Componente y**	**Componente z**
l_1 Vetor de Offset	$TC_CARR1[m]	$TC_CARR2[m]	$TC_CARR3[m]
l_2 Vetor de Offset	$TC_CARR4[m]	$TC_CARR5[m]	$TC_CARR6[m]
v_1 Eixo de rotação	$TC_CARR7[m]	$TC_CARR8[m]	$TC_CARR9[m]
v_2 Eixo de rotação	$TC_CARR10[m]	$TC_CARR11[m]	$TC_CARR12[m]
α_1 Ângulo de rotação α_2 Ângulo de rotação	$TC_CARR13[m] $TC_CARR14[m]		
l_3 Vetor de Offset	$TC_CARR15[m]	$TC_CARR16[m]	$TC_CARR17[m]

Ampliações das variáveis de sistema para porta-ferramentas orientáveis			
Denominação	**Componente x**	**Componente y**	**Componente z**
l_4 Vetor de Offset	$TC_CARR18[m]	$TC_CARR19[m]	$TC_CARR20[m]
Identificador de eixo do eixo de rotação v_1 do eixo de rotação v_2	Identificador de eixo dos eixos de rotação v_1 e v_2 (a ocupação prévia é zero) $TC_CARR21[m] $TC_CARR22[m]		
Tipo de cinemática **T**ool **P**art **M**ixed mode	$TC_CARR23[m]		
	Tipo de cinemática T ->	Tipo de cinemática P ->	Tipo de cinemática M
	Apenas a ferramenta pode ser girada (ocupação prévia)	Apenas a peça de trabalho gira.	A peça de trabalho e a ferramenta giram
Offset do eixo de rotação v_1 do eixo de rotação v_2	Ângulo em graus dos eixos de rotação v_1 e v_2 com a introdução da posição inicial $TC_CARR24[m] $TC_CARR25[m]		
Offset de ângulo do eixo de rotação v_1 do eixo de rotação v_2	Offset dos dentes Hirth em graus dos eixos de rotação v_1 e v_2 $TC_CARR26[m] $TC_CARR27[m]		
Increm. angular v_1 eixo de rotação v_2 eixo de rotação	Incremento dos dentes Hirth em graus dos eixos de rotação v_1 e v_2 $TC_CARR28[m] $TC_CARR29[m]		
Posição mín. do eixo de rotação v_1 do eixo de rotação v_2	Limite de software para posição mínima dos eixos de rotação v_1 e v_2 $TC_CARR30[m] $TC_CARR31[m]		
Posição máx. do eixo de rotação v_1 do eixo de rotação v_2	Limite de software para posição máxima dos eixos de rotação v_1 e v_2 $TC_CARR32[m] $TC_CARR33[m]		
Nome do porta-ferramentas	Ao invés de um número um porta-ferramenta pode receber um nome. $TC_CARR34[m]		

Ampliações das variáveis de sistema para porta-ferramentas orientáveis			
Usuário: Nome de eixo 1 Nome de eixo 2 Identificação **Posição**	Uso intencional durante os ciclos de medição do usuário. $TC_CARR35[m] $TC_CARR36[m] $TC_CARR37[m]		
	$TC_CARR38[m]	$TC_CARR39[m]	$TC_CARR40[m]
Desloc. fino	Parâmetros que podem ser adicionados **aos valores nos parâmetros básicos**.		
l_1 Vetor de Offset	$TC_CARR41[m]	$TC_CARR42[m]	$TC_CARR43[m]
l_2 Vetor de Offset	$TC_CARR44[m]	$TC_CARR45[m]	$TC_CARR46[m]
l_3 Vetor de Offset	$TC_CARR55[m]	$TC_CARR56[m]	$TC_CARR57[m]
l_4 Vetor de Offset	$TC_CARR58[m]	$TC_CARR59[m]	$TC_CARR60[m]
v_1 Eixo de rotação	$TC_CARR64[m]		
v_2 Eixo de rotação	$TC_CARR65[m]		

Indicação

Explicações para os parâmetros

Com "m" é especificado o número do porta-ferramenta a ser descrito.

O `$TC_CARR47` até `$TC_CARR54` assim como o `$TC_CARR61` até `$TC_CARR63` não estão definidos e geram um alarme quando é feita uma tentativa de leitura ou gravação dos mesmos.

Os pontos inicial e final dos vetores de distância podem ser selecionados livremente nos eixos. Os ângulos de rotação $\alpha 1$, $\alpha 2$ em torno dos dois eixos são definidos com 0° no estado inicial do porta-ferramenta. A cinemática de um porta-ferramenta pode ser descrita de várias formas diferentes.

Porta-ferramentas com apenas um ou nenhum eixo de rotação podem ser descritos para um ou ambos eixos de rotação quando zeramos os vetores de direção.
Em um porta-ferramenta sem eixo de rotação atuam os vetores de distância assim como correções adicionais de ferramenta cujos componentes não são afetados durante a mudança dos planos de usinagem (`G17` até `G19`).

Ampliação dos parâmetros

Parâmetros dos eixos de rotação

As variáveis de sistema foram ampliadas com os registros $TC_CARR24[m] até $TC_CARR33[m] e descritas como segue:

O **Offset** dos eixos de rotação v_1, v_2	Alteração da posição do eixo de rotação v_1 ou v_2 na posição inicial do porta-ferramenta orientável.
O **Offset angular/ incremento angular** dos eixos de rotação v_1, v_2	Offset do incremento angular dos dentes Hirth dos eixos de rotação v_1 e v_2. O ângulo programado ou calculado é arredondado para o próximo valor que, no caso do valor inteiro n, resulta do phi = s + n * d.
Posição mínima e máxima dos eixos de rotação v_1, v_2	A posição mínima/máxima do eixo de rotação é o ângulo de limite (limite de software) dos eixos de rotação v1 e v2.

Parâmetros para o usuário

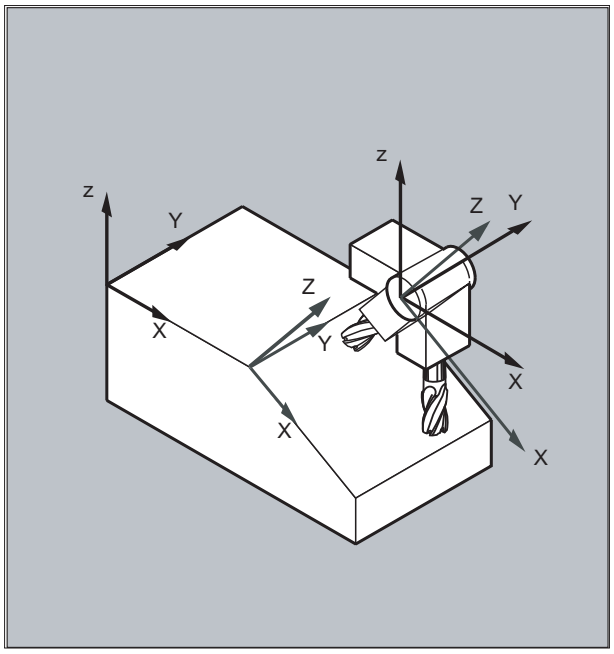
O $TC_CARR34 até $TC_CARR40 contém parâmetros que o usuário tem livre acesso e que, até o SW 6.4, como padrão na NCK, não são mais avaliados ou são irrelevantes.

Parâmetros do deslocamento fino

O $TC_CARR41 até $TC_CARR65 contém parâmetros de deslocamento fino que podem ser adicionados aos valores nos parâmetros básicos. O valor de deslocamento fino atribuído em um parâmetro básico é formado quando se adiciona o valor 40 ao número de parâmetro.

Exemplo

O porta-ferramenta utilizado no exemplo a seguir pode ser completamente descrito por uma rotação em torno do eixo Y.



Código de programa	**Comentário**
N10 $TC_CARR8[1]=1	; Definição do componente Y do primeiro eixo de rotação do porta-ferramenta 1.
N20 $TC_DP1[1,1]=120	; Definição da fresa de topo.
N30 $TC_DP3[1,1]=20	; Definição de uma fresa de topo de 20 mm de comprimento.
N40 $TC_DP6[1,1]=5	; Definição de uma fresa de topo de 5 mm de raio.
N50 ROT Y37	; Definição de Frame com rotação de 37° em torno do eixo Y.
N60 X0 Y0 Z0 F10000	; Aproximar a posição de partida.
N70 G42 CUT2DF TCOFR TCARR=1 T1 D1 X10	; Ajustar a correção do raio, correção do comprimento da ferramenta no Frame girado, selecionar porta-ferramenta 1 e ferramenta 1.
N80 X40	; Executar a usinagem com a rotação de 37°.
N90 Y40	
N100 X0	
N110 Y0	
N120 M30	

Outras informações

Cinemática resolvida

Para máquinas com cinemática resolvida (se gira tanto a ferramenta como a peça) as variáveis de sistema são ampliadas com os registros `$TC_CARR18[m]` até `$TC_CARR23[m]` e descritas como segue:

A mesa giratória de ferramentas é composta pelo(a):

- distância vetorial do segundo eixo rotativo `V2` até o ponto de referência de uma mesa giratória de ferramenta `I4` do terceiro eixo de rotação.

Os eixos rotativos são compostos pelo(a):

- dois identificadores de canal para a referência dos eixos de rotação `V1`e `V2`, cuja posição se acessa eventualmente na definição da orientação do porta-ferramenta orientável.

O tipo de cinemática com um dos valores T, P ou M:

- Tipo de cinemática T: Apenas a ferramenta gira.
- Tipo de cinemática P: Apenas a peça de trabalho gira.
- Tipo de cinemática M: A ferramenta e a peça de trabalho giram.

Deletação dos dados do porta-ferramenta

Com `$TC_CARR1[0]=0` pode-se deletar os dados de todos blocos de dados do porta-ferramenta.

O tipo de cinemática `$TC_CARR23[T]=T` pode ser especificado com uma das três letras maiúsculas ou minúsculas permitidas (T,P,M) e por este motivo não deveria ser deletado.

Alteração dos dados do porta-ferramenta

Cada um dos valores descritos pode ser alterado com a atribuição de um novo valor no programa de peça. Todo caractere diferente de T, P ou M gera um alarme na tentativa de se ativar o porta-ferramenta orientável.

Leitura dos dados do porta-ferramenta

Cada um dos valores descritos pode ser lido no programa de peça através da atribuição em uma variável.

Deslocamentos finos

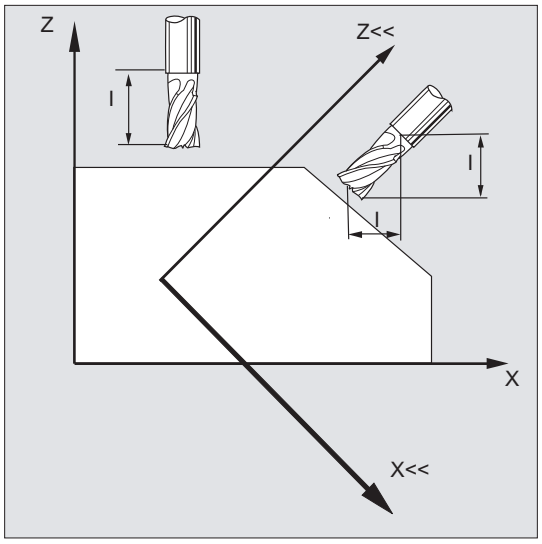
Um valor de deslocamento fino não permitido apenas será detectado quando estiver ativado um porta-ferramenta orientável que tem este tipo de valor e ao mesmo tempo o dado de ajuste SD42974 $SC_TOCARR_FINE_CORRECTION = TRUE.

O valor do deslocamento fino permitido é limitado em um valor máximo permitido através de dados de máquina.

7.9 Correção de comprimento de ferramenta para porta-ferramentas orientáveis (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ)

Função

Com a orientação espacial alterada da ferramenta também são alterados os componentes das mesmas.



Após uma mudança, p. ex. através de ajuste manual ou troca do porta-ferramentas com alinhamento fixo no espaço, se deve determinar novamente os componentes de comprimento da ferramenta. Isto se realiza com os comandos de curso `TCOABS` e `TCOFR`.

Em um porta-ferramenta orientável de um Frame, na seleção de ferramenta com `TCOFRZ`, `TCOFRY` e `TCOFRX`, se pode definir o sentido em que a ferramenta deve apontar.

Sintaxe

```
TCARR=[<m>]
TCOABS
TCOFR
TCOFRZ
TCOFRY
TCOFRX
```

Significado

TCARR=[<m>]:	Solicitar porta-ferramenta com o número "m"
TCOABS:	Calcular os componentes de comprimento da ferramenta a partir da atual orientação do porta-ferramenta
TCOFR:	Definir os componentes de comprimento da ferramenta a partir da orientação do Frame ativo
TCOFRZ:	Porta-ferramenta orientável do Frame ativo cuja ferramenta aponta para o sentido Z
TCOFRY:	Porta-ferramenta orientável do Frame ativo cuja ferramenta aponta para o sentido Y
TCOFRX:	Porta-ferramenta orientável do Frame ativo cuja ferramenta aponta para o sentido X

Outras informações

Correção de comprimento de ferramenta a partir da orientação do suporte (TCOABS)

O TCOABS calcula a correção de comprimento da ferramenta a partir dos atuais ângulos de orientação do porta-ferramenta; armazenada nas variáveis de sistema $TC_CARR13 e $TC_CARR14.

Para definição da cinemática do porta-ferramenta com variáveis de sistema, veja " Cinemática do porta-ferramenta (Página 445) ".

Para recálculo da correção de comprimento da ferramenta na mudança de Frames a ferramenta deve ser selecionada novamente.

Direção da ferramenta a partir do Frame ativo

O porta-ferramenta orientável pode ser ajustado de modo que a ferramenta aponte para as seguintes direções:

- com TCOFR ou TCOFRZ no sentido Z
- com TCOFRY no sentido Y
- com TCOFRX no sentido X

Uma comutação entre TCOFR e TCOABS aciona um recálculo da correção de comprimento da ferramenta.

Solicitar porta-ferramenta (TCARR)

Com TCARR são solicitados os dados geométricos (memória de correções) do número de porta-ferramenta m.

Com m=0 se desseleciona o porta-ferramenta ativo.

Os dados geométricos do porta-ferramenta são ativados apenas após a chamada de uma ferramenta. A ferramenta selecionada permanece ativa durante a mudança de um porta-ferramenta.

Os atuais dados geométricos do porta-ferramenta também podem ser definidos no programa de peça através das respectivas variáveis de sistema.

Recálculo da correção de comprimento da ferramenta (TCOABS) na mudança de Frames

Para recálculo da correção de comprimento da ferramenta na mudança de Frames a ferramenta deve ser selecionada novamente.

Indicação

A orientação de ferramenta deve ser adaptada manual ao Frame ativo.

No recálculo da correção de comprimento da ferramenta também é calculado o ângulo de rotação do porta-ferramenta em um passo intermediário. Visto que porta-ferramentas com dois eixos de rotação normalmente existem em dois pares de ângulo de rotação com os quais a orientação da ferramenta pode ser adaptada ao Frame ativo, os valores dos ângulos de rotação devem, de forma aproximada, corresponder pelo menos aos ângulos de rotação ajustados mecanicamente.

Indicação

Orientação da ferramenta

O comando não pode verificar o ângulo de rotação calculado através da orientação de Frame com o ajustado na máquina.

Se os eixos de rotação do porta-ferramenta construtivamente não puderem alcançar a orientação de ferramenta calculada através da orientação de Frame, então será emitido um alarme.

Não é permitida combinação da correção fina da ferramenta e as funcionalidades para correção de comprimento de ferramenta em porta-ferramentas móveis. Numa tentativa de ativar as duas funções ao mesmo tempo, ocorre uma mensagem de erro.

Com `TOFRAME` é possível definir um Frame com base no sentido de orientação do porta-ferramenta selecionado. Para informações mais detalhadas, veja o capítulo "Frames".

Com a transformação de orientação (transformação de 3, 4 e 5 eixos) ativa, um porta-ferramenta de orientação fora da posição zero pode ser selecionado sem que um alarme seja emitido.

Parâmetro de transferência de ciclos padrão e ciclos de medição

Para o parâmetro de transferência de ciclos padrão e ciclos de medição são aplicadas faixas de valores definidas.

No caso dos valores angulares a faixa de valores está definida da seguinte forma:

- Rotação em torno do 1º eixo geométrico: -180 graus até +180 graus
- Rotação em torno do 2º eixo geométrico: -90 graus até +90 graus
- Rotação em torno do 3º eixo geométrico: -180 graus até +180 graus

Veja o capítulo Frames, "Rotação programável (ROT, AROT, RPL)".

Indicação

Na transferência de valores angulares em um ciclo padrão ou ciclo de medição deve-se observar o seguinte:

Valores menores que a unidade de cálculo do NC devem ser arredondados para zero!

A unidade de cálculo do NC para posições angulares está definida no dado de máquina:

MD10210 $MN_INT_INCR_PER_DEG

7.10 Correção Online de comprimento de ferramenta (TOFFON, TOFFOF)

Função

Através da variável de sistema $AA_TOFF[<n>] os comprimentos efetivos de ferramenta podem ser sobrepostos em tempo real de acordo com os três sentidos da ferramenta.

Como índice <n> são utilizados os três identificadores de eixo geométrico. Com isso se define o número de sentidos de correção ativos através dos eixos geométricos durante o mesmo tempo.

Todas correções podem estar ativas simultaneamente.

A função da correção de comprimento de ferramenta Online pode ser aplicada no(a):

- Transformação de orientação TRAORI
- Porta-ferramenta orientável TCARR

Indicação

A correção de comprimento de ferramenta Online é um **opcional** que deve ser liberado antes. Esta função apenas tem utilidade se trabalhar em conjunto com uma transformação ativa da orientação ou com um porta-ferramenta orientável ativo.

Sintaxe

```
TRAORI
TOFFON(<sentido de correção>[,<valor de Offset>])
WHEN TRUE DO $AA_TOFF[<sentido de correção>]           ; Em ações síncronas.
...
TOFFOF(<sentido de correção>)
```

Para mais explanações sobre programação da correção de comprimento de ferramenta Online em ações síncronas de movimentos, veja o capítulo "Correção Online de comprimento de ferramenta ($AA_TOFF) (Página 608)".

Significado

`TOFFON`: **Ativação** da correção de comprimento de ferramenta Online

`<sentido de correção>`: Sentido de correção (`X`, `Y`, `Z`) onde a correção de comprimento da ferramenta Online deve estar ativo.

`<valor Offset>`: Na ativação se pode especificar um valor de Offset para o respectivo sentido de correção que será imediatamente executado.

`TOFFOF`: **Resetamento** da correção de comprimento de ferramenta Online

Os valores de correção no sentido de correção especificado são resetados e se ativa uma parada de pré-processamento.

Exemplos

Exemplo 1: Seleção da correção de comprimento da ferramenta

Código de programa	Comentário
MD21190 $MC_TOFF_MODE =1	; Os valores absolutos são aproximados.
MD21194 $MC_TOFF_VELO[0] =1000	
MD21196 $MC_TOFF_VELO[1] =1000	
MD21194 $MC_TOFF_VELO[2] =1000	
MD21196 $MC_TOFF_ACCEL[0] =1	
MD21196 $MC_TOFF_ACCEL[1] =1	
MD21196 $MC_TOFF_ACCEL[2] =1	
N5 DEF REAL XOFFSET	
N10 TRAORI(1)	; Transformação ativada.
N20 TOFFON(Z)	; Ativação da correção Online do comprimento da ferramenta para o sentido Z da ferramenta.
N30 WHEN TRUE DO $AA_TOFF[Z]=10 G4 F5	; Para o sentido Z da ferramenta é interpolada uma correção do comprimento da ferramenta de 10
...	
N100 XOFFSET=$AA_TOFF_VAL[X]	; Atribuir atual correção no sentido X.
N120 TOFFON(X,-XOFFSET) G4 F5	; Para o sentido X da ferramenta a correção do comprimento da ferramenta é novamente retornada como 0.

Exemplo 2: Desativação da correção do comprimento da ferramenta

Código de programa	Comentário
N10 TRAORI(1)	; Transformação ativada.
N20 TOFFON(X)	; Ativação da correção Online do comprimento da ferramenta para o sentido X da ferramenta.
N30 WHEN TRUE DO $AA_TOFF[X] = 10 G4 F5	; Para o sentido X da ferramenta é interpolada uma correção do comprimento da ferramenta de 10.
...	
N80 TOFFOF(X)	; O Offset de posição do sentido X da ferramenta é deletado: ...$AA_TOFF[X]=0 Não é movimentado nenhum eixo. Para a atual posição em WCS é adicionado o deslocamento de posição conforme a atual orientação.

Outras informações

Preparação de blocos

Para preparação de blocos no pré-processamento é considerado o atual deslocamento de comprimento da ferramenta ativo no processamento principal. Para permitir um uso extenso das velocidades máximas permitidas dos eixos, é necessário parar a preparação de blocos com uma parada de pré-processamento `STOPRE` enquanto um deslocamento de ferramenta é constituído.

O deslocamento da ferramenta sempre é conhecido no momento do pré-processamento se as correções do comprimento da ferramenta não forem mais alteradas após o início do programa ou se após uma alteração das correções do comprimento da ferramenta não forem mais processados blocos que o buffer IPO pode suportar entre o pré-processamento e o processamento principal.

Variável $AA_TOFF_PREP_DIFF

Na variável `$AA_TOFF_PREP_DIFF[<n>]` é possível consultar a medida da diferença entre a atual correção ativa no interpolador e a correção que estava ativa no momento da preparação de blocos.

Ajustar dados da máquina e dados de ajuste

Para correção de comprimento de ferramenta Online estão disponíveis os seguintes dados de sistema:

- MD20610 $MC_ADD_MOVE_ACCEL_RESERVE (Reserva de aceleração para movimento sobreposto)
- MD21190 $MC_TOFF_MODE

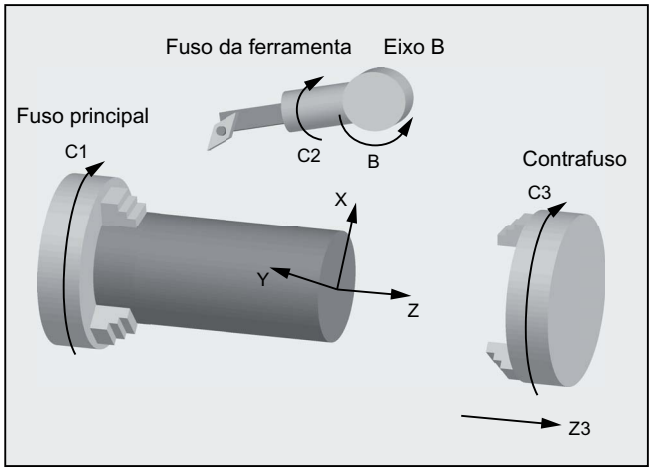
  O conteúdo da variável de sistema `$AA_TOFF[<n>]` é executado ou integrado como valor absoluto.
- MD21194 $MC_TOFF_VELO (Velocidade da correção do comprimento de ferramenta Online)
- MD21196 $MC_TOFF_ACCEL (Aceleração da correção do comprimento de ferramenta Online)
- Dado de ajuste para especificação de valores de limite:
  SD42970 $SC_TOFF_LIMIT (Limite superior do valor de correção do comprimento da ferramenta)

Literatura:
Manual de funções especiais; F2: Transformações múltiplas

7.11 Modificação dos dados de corte para ferramentas orientáveis (CUTMOD)

Função

Com a função "Modificação de dados de corte para ferramentas orientáveis" as relações geométricas alteradas, que resultam da rotação de ferramentas (principalmente ferramentas de tornear, mas também ferramentas de furar e fresar) relativa à peça de trabalho usinada, também podem ser consideradas na correção de ferramenta.



Esquema 7-1 Ferramenta orientável em um torno

Aqui a atual rotação da ferramenta sempre é determinada a partir de um porta-ferramenta orientável atualmente ativo (veja " Correção de comprimento da ferramenta para porta-ferramentas orientáveis (Página 451) ").

A função é ativada com o comando `CUTMOD`.

Sintaxe

```
CUTMOD=<valor>
```

Significado

CUTMOD — Comando para ativar a função "Modificação de dados de corte para ferramentas orientáveis"

<valor> — Ao comando CUTMOD podem ser atribuídos os seguintes valores:

0	A função está desativada. Os valores retornados pelas variáveis $P_AD... são iguais aos parâmetros de ferramenta correspondentes.
> 0	A função é ativada se um porta-ferramenta orientável estiver ativo com o número especificado, isto é, a ativação está associada a um determinado porta-ferramenta orientável. Os valores retornados pelas variáveis de sistema $P_AD..., ao contrário dos parâmetros de ferramenta correspondentes, eventualmente, são modificados em função da rotação ativa. A desativação do porta-ferramenta orientável identificado apenas desativa temporariamente a função, a ativação de outro porta-ferramenta orientável o desativa permanentemente. Por isso que no primeiro caso a função é reativada com a nova seleção do mesmo porta-ferramenta, e no segundo caso, a nova seleção é necessária, mesmo se mais tarde o porta-ferramenta orientável for ativado novamente com o número especificado. A função não é influenciada através do Reset.
-1	A função sempre é ativada se um porta-ferramenta orientável estiver ativo. Na troca do porta-ferramenta ou no caso desse ser desativado e depois novamente ativado, se deve ativar novamente o CUTMOD.
-2	A função sempre é ativada se um porta-ferramenta orientável estiver ativo e seu número for igual ao atual porta-ferramenta orientável ativo. Se nenhum porta-ferramenta orientável estiver ativo, isto tem o mesmo significado que CUTMOD=0. Se um porta-ferramenta orientável estiver ativo, isto tem o mesmo significado que a indicação imediata do atual número de porta-ferramenta.
< -2	Os valores menores que -2 são ignorados, isto é, este caso é tratado como se o CUTMOD não fosse programado. **Nota:** Esta faixa de valores não deve ser utilizada, pois ela está reservada para eventuais futuras ampliações.

Indicação

SD42984 $SC_CUTDIRMOD

A função ativada através do comando CUTMOD substitui a função ativável através do dado de ajuste SD42984 $SC_CUTDIRMOD. Entretanto esta função continua disponível sem ser alterada. Visto que não faz sentido utilizar as duas funções paralelamente, ela somente pode ser ativada se o CUTMOD for igual a zero.

Exemplo

O exemplo a seguir se refere a uma ferramenta com a posição de corte 3 e um porta-ferramenta orientável, que pode girar a ferramenta em torno do eixo B.

Os valores numéricos nos comentários indicam as posições finais do bloco em coordenadas de máquina (MCS) na sequência X, Y e Z.

```
Código de programa                                          Comentário
N10 $TC_DP1[1,1]=500
N20 $TC_DP2[1,1]=3                                          ; Posição de corte
N30 $TC_DP3[1,1]=12
N40 $TC_DP4[1,1]=1
N50 $TC_DP6[1,1]=6
N60 $TC_DP10[1,1]=110                                       ; Ângulo do suporte
N70 $TC_DP11[1,1]=3                                         ; Sentido de corte
N80 $TC_DP24[1,1]=25                                        ; Ângulo livre

N90 $TC_CARR7[2]=0 $TC_CARR8[2]=1 $TC_CARR9[2]=0            ; Eixo B
N100 $TC_CARR10[2]=0 $TC_CARR11[2]=0 $TC_CARR12[2]=1        ; Eixo C
N110 $TC_CARR13[2]=0
N120 $TC_CARR14[2]=0
N130 $TC_CARR21[2]=X
N140 $TC_CARR22[2]=X
N150 $TC_CARR23[2]="M"

N160 TCOABS CUTMOD=0
N170 G18 T1 D1 TCARR=2                                        X           Y           Z
N180 X0 Y0 Z0 F10000                                        ; 12.000      0.000       1.000

N190 $TC_CARR13[2]=30
N200 TCARR=2
N210 X0 Y0 Z0                                               ; 10.892      0.000       -5.134
N220 G42 Z-10                                               ; 8.696       0.000       -17.330
N230 Z-20                                                   ; 8.696       0.000       -21.330
N240 X10                                                    ; 12.696      0.000       -21.330
N250 G40 X20 Z0                                             ; 30.892      0.000       -5.134

N260 CUTMOD=2 X0 Y0 Z0                                      ; 8.696       0.000       -7.330
N270 G42 Z-10                                               ; 8.696       0.000       -17.330
N280 Z-20                                                   ; 8.696       0.000       -21.330
N290 X10                                                    ; 12.696      0.000       -21.330
N300 G40 X20 Z0                                             ; 28.696      0.000       -7.330

N310 M30
```

Explanações:

No bloco N180 se seleciona primeiro a ferramenta com CUTMOD=0 e não o porta-ferramenta orientável girado. Visto que todos vetores de Offset do porta-ferramenta orientável têm valor 0, se aproxima a posição que estiver especificada no $TC_DP3[1,1] e $TC_DP4[1,1] que corresponde aos comprimentos de ferramenta.

No bloco N200 é ativado o porta-ferramenta orientável com uma rotação de 30° em torno do eixo B. Visto que a posição do corte não é alterada devido o CUTMOD=0, prevalece o ponto de referência antigo dos cortes. Por isso que no bloco N210 é aproximada a posição que contém o antigo ponto de referência de corte no ponto zero (isto é, o vetor (1, 12) é girado 30° no plano Z/X).

No bloco N260 atua uma diferença em relação ao bloco N200 CUTMOD=2. Por causa da rotação do porta-ferramenta orientável, a posição de corte modificada passa a ser 8. Disto também resultam posições de eixo divergentes.

Nos blocos N220 e N270 sempre se ativa a correção do raio da ferramenta (WRK). A posição de corte diferente nos dois segmentos do programa não tem nenhuma influência sobre as posições finais dos blocos ativos na WRK, sendo que as posições correspondentes são idênticas. Somente nos blocos de desativação N260 e N300 que atuam novamente as posições de corte diferentes.

Outras informações

Ação dos dados de corte modificados

A posição de corte modificada e o ponto de referência do corte são ativados imediatamente na programação também para uma ferramenta ativa. Uma nova seleção de ferramenta neste caso não é necessária.

Influência do plano de trabalho ativo

Para a definição da posição de corte, sentido de corte e ângulo de suporte ou ângulo livre modificados é determinante se considerar o corte no respectivo plano ativado (G17 - G19).

Entretanto, se o dado de ajuste SD42940 $SC_TOOL_LENGTH_CONST (troca dos componentes de comprimento da ferramenta na mudança de planos) contém um valor válido diferente de zero (positivo ou negativo 17, 18 ou 19), então este conteúdo determinará o plano que as grandezas relevantes devem ser consideradas.

Variáveis de sistema

Estão disponíveis as seguintes variáveis de sistema:

Variáveis de sistema	Significado
$P_CUTMOD_ANG / $AC_CUTMOD_ANG	Retorna o ângulo (não arredondado) no plano de usinagem ativo, que foi definido para a modificação dos dados de corte (posição de corte, sentido de corte, ângulo e ângulo do suporte) com as funções CUTMOD e $SC_CUTDIRMOD ativadas. O $P_CUTMOD_ANG se refere ao atual estado no pré-processamento, e o $AC_CUTMOD_ANG ao atual bloco de processamento principal.
$P_CUTMOD / $AC_CUTMOD	Lê o atual valor válido que foi programado por último com o comando CUTMOD (número do porta-ferramenta que se deve ativar a modificação de dados de corte). Era o último valor CUTMOD = -2 programado (ativação com o atual porta-ferramenta orientável ativo), no $P_CUTMOD não se retorna o valor -2, mas o número do porta-ferramenta orientável ativo no momento da programação. O $P_CUTMOD se refere ao atual estado no pré-processamento, e o $AC_CUTMOD ao atual bloco de processamento principal.
$P_CUT_INV / $AC_CUT_INV	Retorna o valor TRUE se a ferramenta for girada de modo que o sentido de giro do fuso seja invertido. Para isso devem ser preenchidas as seguintes quatro condições no bloco relacionado à respectiva operação de leitura: 1. Uma ferramenta de tornear ou de retificar está ativa (tipos de ferramenta 400 até 599 e / ou SD42950 $SC_TOOL_LENGTH_TYPE = 2). 2. O controle dos cortes foi ativado com o comando de linguagem CUTMOD. 3. Um porta-ferramenta orientável que foi identificado pelo valor numérico do CUTMOD está ativo. 4. O porta-ferramentas orientável gira a ferramenta em torno de um eixo no plano de usinagem (tipicamente o eixo C) de modo que a normal resultante do corte da ferramenta esteja girada mais que 90° (tipicamente 180°) em relação à posição de saída. Se apenas uma das quatro condições mencionadas não for preenchida, então o conteúdo da variável é FALSE. Para ferramentas cuja posição de corte não está definida, o valor da variável é sempre FALSE. O $P_CUT_INV se refere ao atual estado no pré-processamento e o $AC_CUT_INV ao atual bloco de processamento principal.

Todas variáveis de processamento principal ($AC_CUTMOD_ANG, $AC_CUTMOD e $AC_CUT_INV) podem ser lidas em ações sincronizadas. Um acesso de leitura a partir do pré-processamento gera uma parada de pré-processamento.

Dados de corte modificados:

Se uma rotação de ferramenta estiver ativa, os dados modificados são disponibilizados nas seguintes variáveis de sistema:

Variável de sistema	Significado
$P_AD[2]	Posição de corte
$P_AD[10]	Ângulo do suporte
$P_AD[11]	Sentido de corte
$P_AD[24]	Ângulo livre

Indicação

Os dados sempre são modificados em relação aos parâmetros de ferramentas correspondentes ($TC_DP2[..., ...] etc.) quando a função "Modificação de dados de corte para ferramentas orientáveis" for ativada com o comando `CUTMOD` e um porta-ferramenta orientável que executa a rotação de uma ferramenta estiver ativo.

Literatura

Para mais informações sobre a função "Modificação de dados de corte para ferramentas orientáveis", veja:

Manual de funções básicas; Correção de ferramenta (W1)

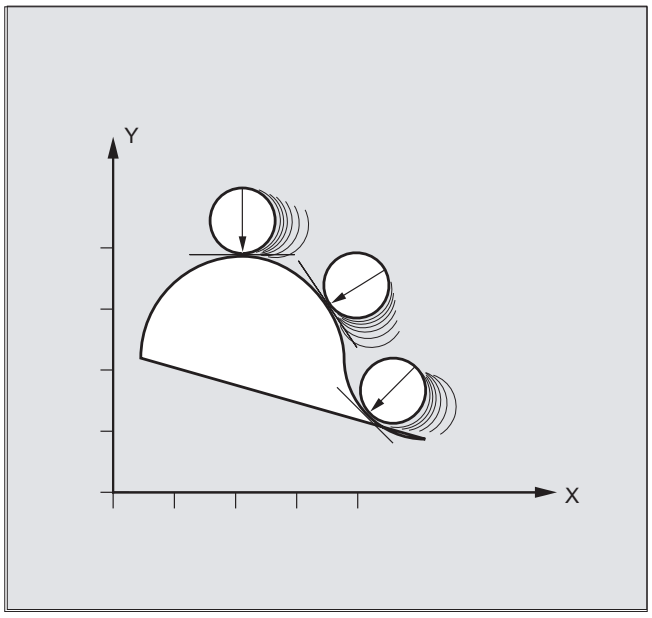
Comportamento de percurso

8

8.1 Controle tangencial (TANG, TANGON, TANGOF, TLIFT, TANGDEL)

Função

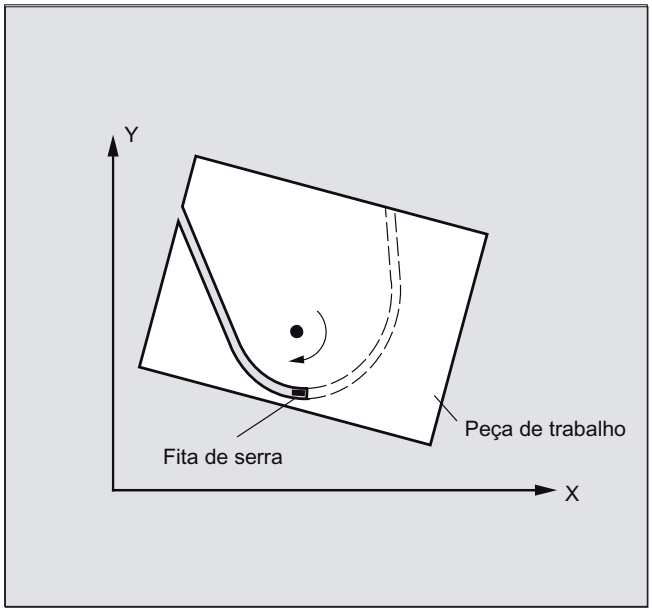
O eixo escravo é acompanhado conforme a tangente no percurso definido pelos eixos mestres. Com isso uma ferramenta pode ser alinhada paralela ao contorno. Através do ângulo programado na instrução `TANGON` a ferramenta pode ser posicionada relativa à tangente.



Aplicação

Por exemplo, o controle tangencial pode ser aplicado para:

- Posicionamento tangencial de uma ferramenta orientável na estampagem
- Acompanhamento do alinhamento da peça de trabalho com uma serra tipo fita (veja a próxima figura)
- Posicionamento de um dressador em um rebolo
- Posicionamento de um disco de corte para processar vidro e papel
- Alimentação tangencial de um fio para soldagem com 5 eixos



Sintaxe

Definição do acompanhamento tangencial:
`TANG(<EixoE>,<EixoM1>,<EixoM2>,<fator de acoplamento>,<SC>,<Opt>)`

Ativação do controle tangencial:
`TANGON(<EixoE>,<ângulo>,<dist>,<tolerância angular>)`

Desativação do controle tangencial:
`TANGOF(<EixoE>)`

Ativação da função "Inserir bloco intermediário nos cantos do contorno":
`TLIFT(<EixoE>)`

A instrução `TLIFT` é especificada logo após a atribuição de eixos `TANG(…)`.

Desativação da função "Inserir bloco intermediário nos cantos do contorno":

Repetição da instrução `TANG(...)` sem a sequência do `TLIFT(<EixoE>)`.

Deletação da definição de um acompanhamento tangencial:
`TANGDEL(<EixoE>)`

Um acompanhamento tangencial definido pelo usuário deve ser deletado quando for definido um novo acompanhamento tangencial de mesmo eixo escravo na chamada do processamento do `TANG`. Uma deletação apenas é possível se o acoplamento for desativado com `TANGOF(<EixoE>)`.

Significado

`TANG`:	Instrução a ser preparada para definição de um acompanhamento tangencial
`TANGON`:	Ativação do controle tangencial para o eixo escravo especificado

`TANGOF`: Desativação do controle tangencial para o eixo escravo especificado

`TLIFT`: Ativação da função "Inserir bloco intermediário nos cantos do contorno"

`TANGDEL`: Deletação da definição de um acompanhamento tangencial

`<EixoE>`: Eixo escravo: Eixo rotativo adicional acompanhado tangencialmente

`<EixoM1>,<EixoM2>`: Eixos mestres: Eixos de percurso a partir dos quais se define a tangente para o acompanhamento

`<fator de acoplamento>`: Fator de acoplamento: Relação entre a alteração angular da tangente e o eixo acompanhado

Pré-ajuste: 1

Nota:
Um fator de acoplamento de 1 não precisa ser programado de forma explícita.

`<SC>`: Letras de identificação para sistema de coordenadas

`"B"`: Sistema de coordenadas básico (pré-ajuste)

Nota:
<`SC> = "B"` não precisa ser programado de forma explícita.

`"W"`: Sistema de coordenadas da peça de trabalho (não disponível)

`<Opt>`: Otimização

`"S"`: Standard (pré-ajuste)

Nota:
<`Opt> = "S"` não precisa ser programado de forma explícita.

`"P"`: Adaptação automática do tempo gasto do eixo tangencial e do contorno

Nota:
Com **<**`Opt> = "P"` a dinâmica do eixo escravo não será considerada na limitação de velocidade dos eixos mestres. Este ajuste é recomendado principalmente no emprego das transformações cinemáticas.

`<ângulo>`: Ângulo de deslocamento do eixo escravo

`<Dist>`: Curso de suavização do eixo escravo (necessário se `<Opt>` **=** `"P"`)

`<tolerância angular>`: Tolerância angular do eixo escravo (opcional; avaliado somente se `<Opt>` **=** `"P"`)

Nota:
Os parâmetros (`<Dist>` e `<tolerância angular>`) limitam os erros entre o eixo acompanhado e a tangente dos eixos mestres de forma controlada.

Exemplos

Exemplo 1: Definição e ativação do acompanhamento tangencial

Código de programa	Comentário
N10 TANG(C,X,Y,1,"B","P")	; Definição de um acompanhamento tangencial: O eixo rotativo C deve acompanhar os eixos geométricos X e Y.
N20 TANGON(C,90)	; O eixo C é o eixo escravo. Ele é girado em uma posição de 90° em relação à tangente do percurso em cada movimento dos eixos de percurso.
...	

Indicação

Programação simplificada

`TANG(C,X,Y,1,"B","P")` pode ser programado de forma simplificada como `TANG(C,X,Y,,,"P")`.

Exemplo 2: Mudança de planos

Código de programa	Comentário
N10 TANG(A,X,Y,1)	; 1ª definição do acompanhamento tangencial.
N20 TANGON(A)	; Ativação do acoplamento.
N30 X10 Y20	; Raio
...	
N80 TANGOF(A)	; Desativação do 1° acoplamento.
N90 TANGDEL(A)	; Deletação da 1ª definição.
...	
TANG(A,X,Z)	; 2ª definição do acompanhamento tangencial.
TANGON(A)	; Ativação do novo acoplamento.
...	
N200 M30	

Exemplo 3: Comutação de eixos geométricos e TANGDEL

Não é gerado nenhum alarme.

```
Código de programa                  Comentário
N10 GEOAX(2,Y1)                     ; Y1 é o eixo geométrico 2.
N20 TANG(A,X,Y)                     ; 1ª definição do acompanhamento tangencial.
N30 TANGON(A,90)                    ; Ativação do acompanhamento com Y1
N40 G2 F8000 X0 Y0 I0 J50
N50 TANGOF(A)                       ; Desativação do acompanhamento com Y1.
N60 TANGDEL(A)                      ; Deletação da 1ª definição.
N70 GEOAX(2, Y2)                    ; Y2 é o eixo geométrico 2.
N80 TANG(A,X,Y)                     ; 2ª definição do acompanhamento tangencial.
N90 TANGON(A,90)                    ; Ativação do acompanhamento com Y2.
...
```

Exemplo 4: Acompanhamento tangencial com otimização automática

Y1 é o eixo geométrico 2.

```
Código de programa                                  Comentário
...
N80 G0 C0
N100 F=50000
N110 G1 X1000 Y500
N120 TRAORI
N130 G642                                           ; Suavização com preservação do desvio
                                                      de percurso máximo permitido.
N171 TRANS X50 Y50
N180 TANG(C,X,Y,1,,"P")                             ; Definição do acompanhamento tangencial
                                                      com otimização automática da
                                                      velocidade de percurso.
N190 TANGON(C,0,5.0,2.0)                            ; Ativação do acompanhamento tangencial
                                                      com otimização automática: Percurso de
                                                      suavização de 5 mm, tolerância angular
                                                      de 2 graus.
N210 G1 X1310 Y500
N215 G1 X1420 Y500
N220 G3 X1500 Y580 I=AC(1420) J=AC(580)
N230 G1 X1500 Y760
N240 G3 X1360 Y900 I=AC(1360) J=AC(760)
N250 G1 X1000 Y900
N280 TANGOF(C)
N290 TRAFOOF
N300 M02
```

Outras informações

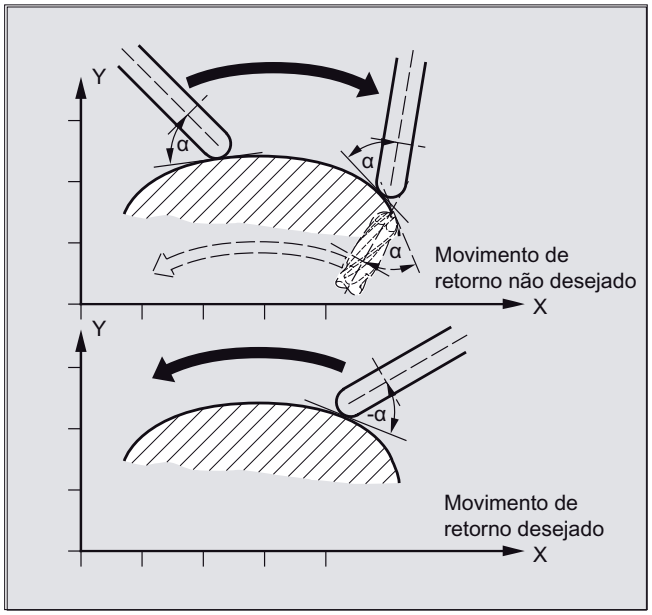
Definir eixos escravo e eixos mestres

A definição de eixos escravos e eixos mestres é realizada com `TANG`.

Um fator de acoplamento estabelece a relação entre uma alteração angular da tangente e o eixo acompanhado. Seu valor normalmente é 1 (pré-ajuste).

Ângulo limite por limite da área de trabalho

Para os movimentos de percurso conduzidos alternadamente, a tangente gira 180° no ponto de reversão, o alinhamento do eixo escravo altera-se de acordo. Normalmente este procedimento não faz muito sentido: O movimento de retorno deve ser percorrido com o mesmo ângulo de deslocamento negativo como no movimento de ida:



Para isso a área de trabalho do eixo escravo deve ser limitada (`G25`, `G26`). O limite da área de trabalho deve estar ativo no momento da reversão do percurso (`WALIMON`). Se o ângulo de deslocamento estiver fora dos limites da área de trabalho será feita a tentativa com o ângulo de deslocamento negativo de retornar à área de trabalho permitida.

Inserção de bloco intermediário nos cantos de contorno (TLIFT)

Em um canto do contorno é alterada a tangente e com isso de forma brusca a posição nominal do eixo acompanhado. O eixo normalmente tenta compensar este salto com sua velocidade máxima possível. Mas neste caso, após o canto resulta um desvio em relação à posição tangencial em um determinado trecho do contorno. Se por motivos tecnológicos isto não for tolerável, pelo comando pode-se fazer uma parada no canto através da instrução `TLIFT` e então girar o eixo acompanhado para o novo sentido da tangente com um bloco intermediário automaticamente gerado.

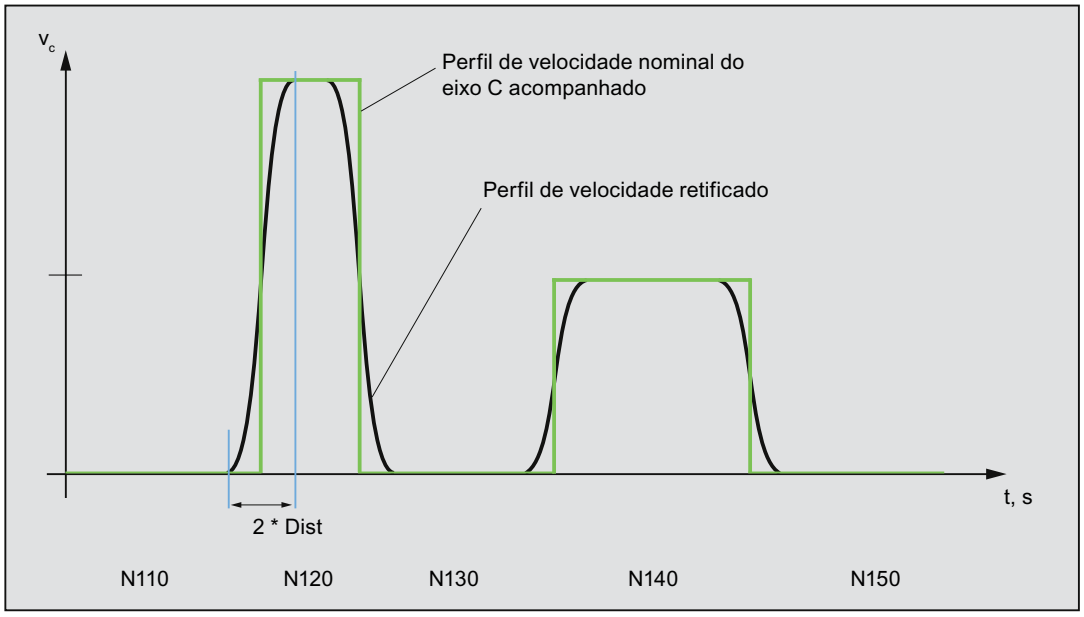
A rotação é realizado com o eixo de percurso programado, se o eixo acompanhado foi movimentado uma vez como eixo de percurso. Neste caso, com a função `TFGREF[<eixo>]=0.001` é possível alcançar uma velocidade máxima do eixo acompanhado.

Se anteriormente o eixo acompanhado não foi movimentado como eixo de percurso, então este eixo é movimentado com eixo de posicionamento. A velocidade está em função da velocidade de posicionamento armazenada no dado da máquina.

A rotação é realizada com a velocidade máxima do eixo acompanhado.

Opção de otimização

Se a otimização automática estiver selecionada (`<Opt>="P"`) e se para o eixo escravo foram especificados os parâmetros de percurso de suavização (`<Dist>`) e tolerância angular (`<tolerância angular>`), então ,no acompanhamento tangencial, ocorre a suavização dos saltos de velocidade do eixo escravo em função de saltos ocorridos no contorno do eixo mestre. Aqui o eixo escravo é controlado antecipadamente (veja o diagrama) para manter o desvio o menor possível.



Definir alteração angular

A alteração angular a partir da qual se insere um bloco intermediário automático é definida através do seguinte dado de máquina:

MD37400 $MA_EPS_TLIFT_TANG_STEP (Ângulo tangencial para detecção de cantos)

Controle sobre transformações

A posição do eixo rotativo acompanhado pode ser o valor de entrada para uma transformação.

Posicionamento explícito do eixo escravo

Se um eixo escravo que acompanha um de seus eixos mestres for posicionado explicitamente, então a posição será adicionada ao ângulo de deslocamento programado.

São permitidas todas as especificações de percurso (movimentos de percurso e de posicionamento).

Estado do acoplamento

No programa de peça NC é possível consultar o estado do acoplamento com a variável de sistema $AA_COUP_ACT[<eixo>]:

Valor	Significado
0	Nenhum acoplamento ativo
1,2,3	Acompanhamento tangencial ativo

8.2 Característica de avanço (FNORM, FLIN, FCUB, FPO)

Função

Para uma especificação flexível da característica de avanço a programação do avanço conforme DIN 66025 é ampliada em características lineares e cúbicas.

As características cúbicas podem ser programadas diretamente ou como Splines interpoladoras. Com isso pode-se programar características de velocidade suaves e contínuas, em função da curvatura da peça de trabalho usinada.

Estas características de velocidade permitem mudanças de aceleração isentas de solavancos tendo como resultado uma usinagem uniforme das superfícies das peças de trabalho.

Sintaxe

```
F… FNORM
F… FLIN
F… FCUB
F=FPO(…,…,…)
```

Significado

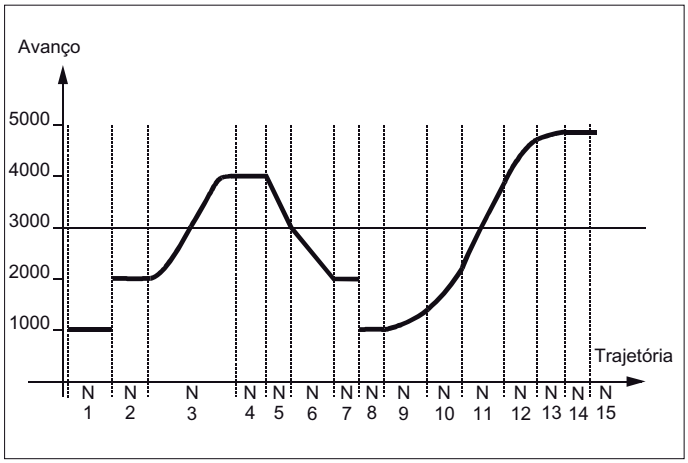
`FNORM`	Ajuste básico. O valor do avanço é especificado através do percurso do bloco e depois ele vale como valor modal.
`FLIN`	Perfil de velocidade de percurso **linear:** O valor de avanço é executado linearmente pelo percurso a partir do atual valor no início do bloco até o fim do bloco e depois ele vale como valor modal. Esta característica pode ser combinada com G93 e G94.
`FCUB`	Perfil da velocidade de percurso **cúbica:** Os valores F programados por blocos são, em função do ponto final do bloco, ligados por uma Spline. A Spline inicia e finda tangencialmente com os avanços anterior e posterior definidos e tem efeito com G93 e G94. Se faltar o endereço F em um bloco, então neste caso se utiliza o último valor F programado.
`F=FPO…`	Perfil da velocidade de percurso **através de polinômio:** O endereço F define a característica de avanço através de um polinômio a partir do atual valor até o fim do bloco. Depois o valor final vale como valor modal.

Otimização de avanço em trechos de percurso curvados

O polinômio de avanço `F=FPO` e a Spline de avanço `FCUB` sempre deveriam ser afastados com velocidade de corte constante `CFC`. Com isso é possível produzir um perfil de avanço nominal com aceleração isenta de solavancos.

Exemplo: Diferentes perfis de avanço

Neste exemplo temos a programação e a representação gráfica de diversos perfis de avanço.

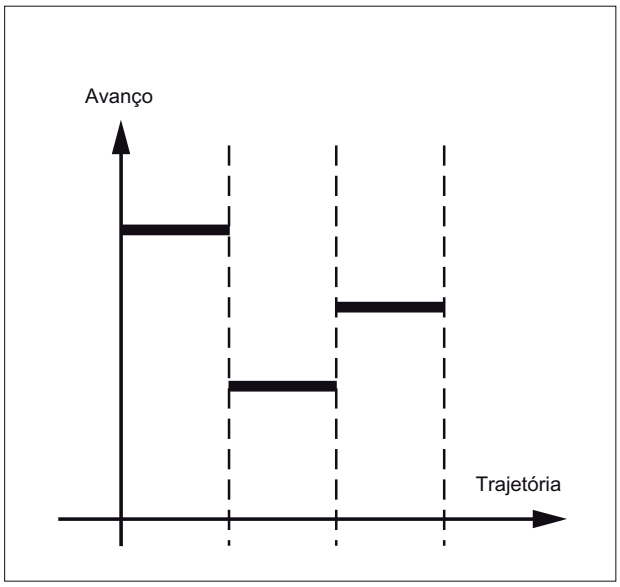


Código de programa	**Comentário**
N1 F1000 FNORM G1 X8 G91 G64	; Perfil de avanço constante, especificação incremental
N2 F2000 X7	; Mudança brusca de velocidade nominal
N3 F=FPO(4000, 6000, -4000)	; Perfil de avanço através de polinômio com avanço 4000 no fim do bloco
N4 X6	; O avanço de polinômio 4000 vale como valor modal
N5 F3000 FLIN X5	; Perfil de avanço linear
N6 F2000 X8	; Perfil de avanço linear
N7 X5	O avanço linear vale como valor modal
N8 F1000 FNORM X5	; Perfil constante de avanço com mudança brusca de aceleração
N9 F1400 FCUB X8	; Todos valores F programados por blocos a seguir serão ligados com Splines
N10 F2200 X6	
N11 F3900 X7	
N12 F4600 X7	
N13 F4900 X5	; Desativar perfil de Spline
N14 FNORM X5	
N15 X20	

FNORM

O endereço de avanço F define o avanço de trajetória como valor constante conforme DIN 66025.

Para mais informações sobre este assunto, veja no Manual de programação "Fundamentos".

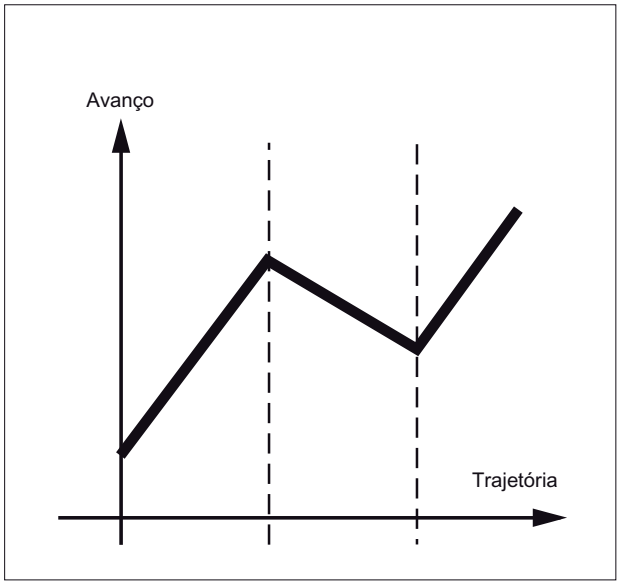


FLIN

A característica de avanço é executada linearmente do atual valor de avanço o valor F programado até atingir o final do bloco.

Exemplo:

```
N30 F1400 FLIN X50
```

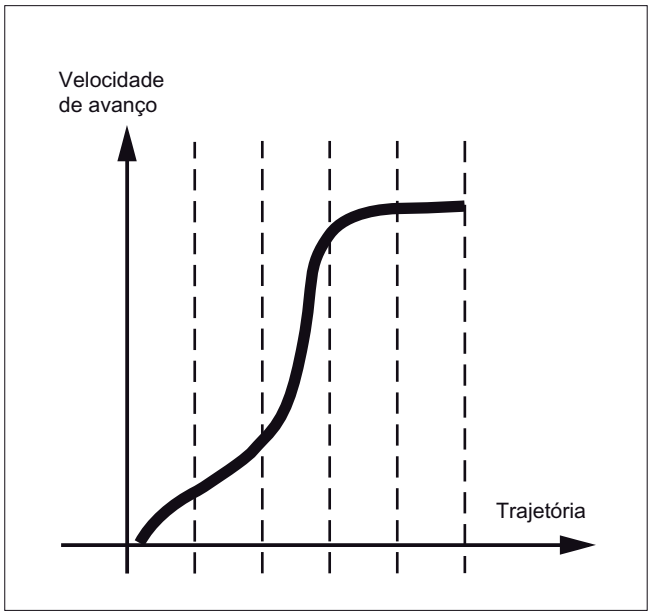


FCUB

O avanço é executado de acordo com a característica cúbica do atual valor de avanço para o valor F até o final do bloco. O comando liga com Splines todos valores de avanço programados por bloco com FCUB ativo. Aqui os valores de avanço servem como pontos de apoio para o cálculo da interpolação de Spline.

Exemplo:

```
N50 F1400 FCUB X50
N60 F2000 X47
N70 F3800 X52
```



F=FPO(…,…,…)

A característica de avanço é programada diretamente através de um polinômio. A especificação do coeficiente de polinômio é realizado de forma similar à interpolação de polinômios.

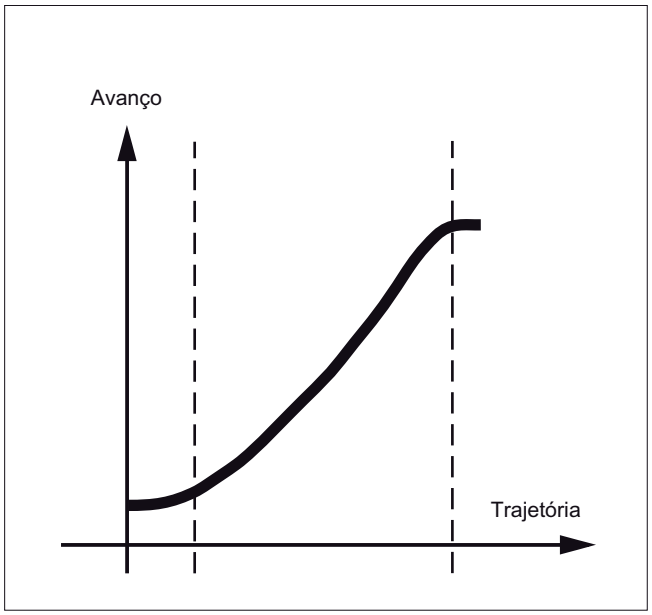
Exemplo:

```
F=FPO(endfeed, quadf, cubf)
```

endfeed, quadf e cubf são variáveis que são definidas antes.

endfeed:	Avanço no fim do bloco
quadf:	Coeficiente quadrático de polinômios
cubf:	Coeficiente cúbico de polinômios

Com `FCUB` ativo a Spline liga tangencialmente à característica definida por `FPO`, no início do bloco e no fim do bloco.



Condições gerais

Independentemente da característica de avanço programada, aplicam-se as funções de programação do comportamento de percurso.

A característica de avanço programável sempre é aplicada de forma absoluta, independentemente de `G90` ou `G91`.

A característica de avanço FLIN e FCUB atua com

`G93` e `G94`.

`FLIN` e `FCUB` **não atua com**

`G95`, `G96/G961` e `G97/G971`.

Compressor ativo COMPON

Com o compressor `COMPON` ativo em um agrupamento podem ser aplicados vários blocos em um segmento de Spline:

FNORM:

Para o segmento de Spline se aplica a palavra F do último bloco correspondente.

FLIN:

Para o segmento de Spline se aplica a palavra F do último bloco correspondente.
O valor F programado vale até o fim do segmento e depois ele é aproximado linearmente.

FCUB:

A Spline de avanço tem um desvio máximo equivalente ao valor definido no dado de máquina `C $MC_COMPRESS_VELO_TOL` referente aos pontos finais programados.

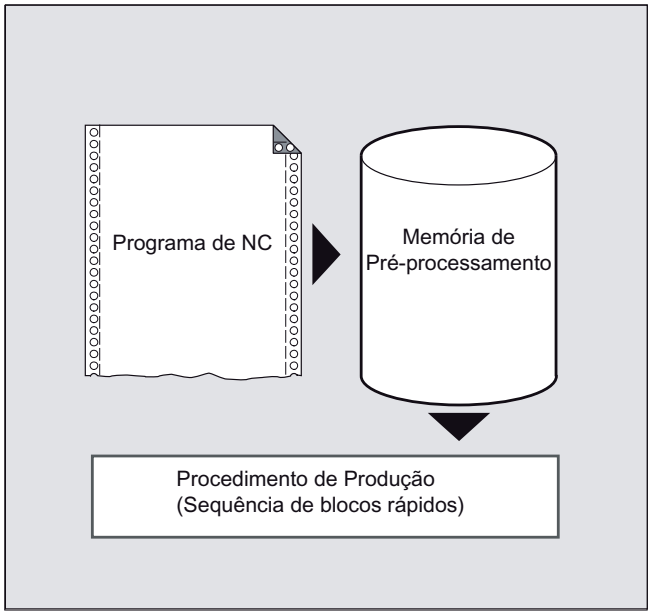
F=FPO(...,...,...)

Estes blocos não são comprimidos.

8.3 Execução do programa com memória de pré-processamento (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE)

Função

Dependendo do nível de expansão, o comando dispõe através de uma determinada quantidade da assim chamada memória de pré-processamento na qual são armazenados os blocos antes deles serem executados, depois são disparadas seqüências rápidas de blocos durante a execução da usinagem. Com isso se pode percorrer pequenos cursos em alta velocidade. Assim que o tempo restante do comando permitir, a memória de pré-processamento é continuamente alimentada.



Marcar o segmento de processamento

O segmento de processamento que deve ser armazenado temporariamente na memória de pré-processamento, é marcado no início com STOPFIFO e no fim com STARTFIFO. A execução dos blocos preparados e armazenados temporariamente somente inicia após o comando STARTFIFO ou quando a memória de pré-processamento estiver cheia.

Controle automático de memória de pré-processamento

O controle automático de memória de pré-processamento é chamada com o comando FIFOCTRL. Em princípio o FIFOCTRL atua da mesma forma como o STOPFIFO. Em cada programação espera-se até a memória de pré-processamento estar cheia, para então ser iniciada a execução. De modo contrário, a diferença fica no comportamento do esvaziamento da memória de pré-processamento. com o FIFOCTRL existe uma redução crescente da velocidade de percurso a partir de um nível de enchimento de 2/3, para evitar um esvaziamento total e a desaceleração até a imobilização total (parada).

Parada de pré-processamento

A preparação e o armazenamento temporário dos blocos são parados, se no bloco estiver programado o comando STOPRE. O bloco seguinte somente será executado se todos blocos anteriormente pré-processados e armazenados foram totalmente executados. O bloco anterior é parado na parada exata (como o G9).

Sintaxe

Tabelas 8- 1 Marcação do segmento de processamento:

```
STOPFIFO
...
STARTFIFO
```

Tabelas 8- 2 Controle automático de memória de pré-processamento:

```
...
FIFOCTRL
...
```

Tabelas 8- 3 Parada de pré-processamento:

```
...
STOPRE
...
```

Indicação

Os comandos STOPFIFO, STARTFIFO, FIFOCTRL e STOPRE devem ser programados em um bloco próprio.

Significado

STOPFIFO: O STOPFIFO marca o início de um segmento de processamento, que deve ser ser armazenado temporariamente na memória de pré-processamento. Com o STOPFIFO o processamento é parado e a memória de pré-processamento é abastecida até:

- STARTFIFO ou STOPRE ser identificado

  ou
- a memória de pré-processamento estiver cheia

  ou
- o fim do programa for alcançado.

STARTFIFO: Com o STARTFIFO é iniciada uma execução rápida do segmento de processamento, paralelamente a isso ocorre o abastecimento da memória de pré-processamento

FIFOCTRL: Ativação do controle automático de memória de pré-processamento

STOPRE: Parada do pré-processamento

Indicação

O abastecimento da memória de pré-processamento não é executado ou é interrompido, se o segmento de processamento conter comandos que forçam uma operação sem memória temporária (aproximação do ponto de referência, funções de medição, …).

Indicação

Ao acessar dados de estado da máquina ($SA...) o comando numérico gera uma parada interna do pré-processamento.

⚠ CUIDADO

Com a compensação de ferramenta ativada e em interpolações de Spline não deve ser programado nenhum STOPRE, pois as seqüências de blocos associadas seriam interrompidas.

Exemplo: Parada do pré-processamento

```
Código de programa                          Comentário
...
N30 MEAW=1 G1 F1000 X100 Y100 Z50           ; Bloco de medição com apalpador da
                                              primeira entrada de medição e
                                              interpolação de retas.
N40 STOPRE                                  ; Parada de pré-processamento.
...
```

8.4 Segmentos de programa interrompidos condicionalmente (DELAYFSTON, DELAYFSTOF)

Função

Os segmentos de programa de peça que podem ser interrompidos condicionalmente são chamados de áreas Stop-Delay. Dentro de determinados segmentos do programa não deve ocorrer nenhuma **parada** e o **avanço** também não deve ser alterado. Basicamente apenas pequenos segmentos de programa, por exemplo os usados na produção de uma rosca, devem ser protegidos de quase todos eventos de parada. Uma eventual parada apenas tem efeito depois que o segmento de programa for executado até o fim.

Sintaxe

```
DELAYFSTON
DELAYFSTOF
```

Os comandos estão isolados em uma linha do programa de peça.

Os dois comandos somente estão em programas de peça, mas não são permitidos em ações sincronizadas.

Significado

`DELAYFSTON`	Definir o início de uma área onde se aplica um retardamento nas paradas "suaves" até o fim da área Stop-Delay ser alcançado.
`DELAYFSTOF`	Definir o fim de uma área Stop-Delay

Indicação

No dado de máquina 11550: STOP_MODE_MASK Bit 0 = 0 (default) se define implicitamente uma área Stop-Delay, se o `G331/G332` estiver ativo e for programado um movimento de percurso ou `G4`.

Exemplo: Eventos de parada

Na área Stop-Delay é ignorada uma mudança do **avanço e bloqueio de avanço**. Estes apenas surtem efeito após a área Stop-Delay.

Os eventos de parada são distinguidos em:

Eventos "suaves" de parada	Reação: delayed
Eventos "bruscos" de parada	Reação: immediate

Seleção de eventos de parada próprios que pelo menos param em curto intervalo:

Nome do evento	Reação	Parâmetro de interrupção
RESET	immediate	NST: DB21,… DBX7.7 e DB11, … DBX20.7
PROG_END	Alarme 16954	Prog-NC: M30
INTERRUPT	delayed	NST: FC-9 e ASUP DB10, ... DBB1
SINGLEBLOCKSTOP	delayed	Modo de operação bloco a bloco ativado na área Stop-Delay: O NC pára no fim do 1º bloco fora da área Stop-Delay. Bloco a bloco selecionado antes da área Stop-Delay: NST: "Parada NC no limite do bloco" DB21, ... DBX7.2
STOPPROG	delayed	NST: DB21,… DBX7.3 e DB11, … DBX20.5
PROG_STOP	Alarme 16954	Prog-NC: M0 e M1
WAITM	Alarme 16954	Prog-NC: WAITM
WAITE	Alarme 16954	Prog-NC: WAITE
STOP_ALARM	immediate	Alarme: Projeção de alarme STOPBYALARM
RETREAT_MOVE_THREAD	Alarme 16954	Prog-NC: Alarme 16954 com LFON (o Stopp & Fastlift no G33 não é possível)
WAITMC	Alarme 16954	Prog-NC: WAITMC
NEWCONF_PREP_STOP	Alarme 16954	Prog-NC: NEWCONF
SYSTEM_SHUTDOWN	immediate	Encerramento de sistema no 840Di
ESR	delayed	Parada e retrocesso ampliados
EXT_ZERO_POINT	delayed	Deslocamento externo de ponto zero
STOPRUN	Alarme 16955	BTSS: PI "_N_FINDST" STOPRUN

Explanação das reações

immediate (evento "brusco" de parada)	Para imediata também na área Stop-Delay
delayed (evento "suave" de parada)	A parada (mesmo uma parada curta) somente é realizada após a área Stop-Delay.
Alarme 16954	O programa é cancelado, pois foram utilizados comandos de programa não permitidos na área Stop-Delay.
Alarme 16955	O programa é continuado, na área Stop-Delay ocorreu uma ação não permitida.
Alarme 16957	A área do programa (área Stop-Delay), que também está compreendida pelo DELAYFSTON e DELAYFSTOF, não pôde ser ativada. Com isso toda parada atua imediatamente na área e não com retardamento.

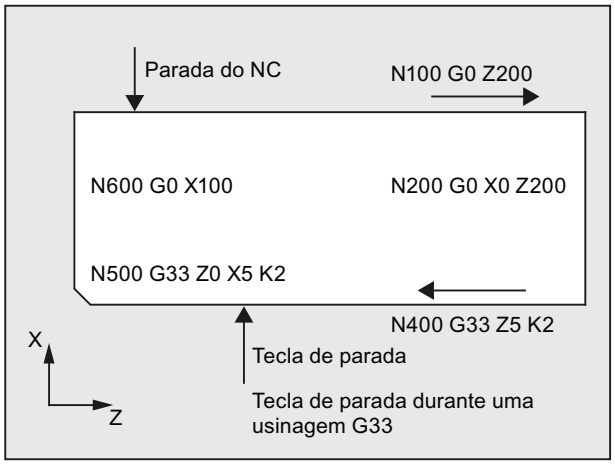
Para um resumo das outras reações nos eventos de parada, veja:
Literatura:
Manual de funções básicas; BAG, Canal, Modo de operação do programa, (K1), capítulo "Controle e efeito sobre eventos de parada"

Exemplo: Aninhamento de áreas Stop-Delay em dois níveis de programa

```
Código de programa          Comentário
N10010 DELAYFSTON()         ; Blocos com N10xxx do nível de programa 1.
N10020 R1 = R1 + 1
N10030 G4 F1                ; É iniciada a área Stop-Delay.
...
N10040 Subrotina2
...
...                         ; Interpretação da subrotina 2.
N20010 DELAYFSTON()         ; Inativo, início repetido, 2° nível.
...
N20020 DELAYFSTOF()         ; Inativo, fim em outro nível.
N20030 RET
N10050 DELAYFSTOF()         ; Área Stop-Delay, fim no mesmo nível.
...
N10060 R2 = R2 + 2
N10070 G4 F1                ; A área Stop-Delay finaliza. A partir de agora as
                              paradas tem efeito imediato.
```

Exemplo: Recorte do programa

Em um loop é repetido o seguinte bloco de programa:



Na figura está visível que o usuário pressiona um "Stop" na área Stop-Delay, e o NC inicia o processo de frenagem fora da área Stop-Delay, isto é, no bloco `N100`. Com isso o NC é parado na área anterior do `N100`.

Código de programa	Comentário
...	
N99 MY_LOOP:	
N100 G0 Z200	
N200 G0 X0 Z200	
N300 DELAYFSTON()	
N400 G33 Z5 K2 M3 S1000	
N500 G33 Z0 X5 K3	
N600 G0 X100	
N700 DELAYFSTOF()	
N800 GOTOB MY_LOOP	

Para detalhes sobre localização de blocos do tipo SERUPRO e avanços em conjunto com o avanço G331/G332 no rosqueamento sem mandril de compensação, veja:
Literatura:
Manual de funções básicas; BAG, Canal, Modo de operação do programa (K1)
Manual de funções básicas; Avanços (V1)

Vantagens da área Stop-Delay

Um segmento do programa é executado sem interrupção de velocidade.

Se o usuário cancela o programa com RESET depois da parada, então o bloco de programa cancelado está após a área protegida. Este bloco de programa é próprio para destino de busca para uma localização de blocos subseqüente.

Enquanto uma área Stop-Delay for executada, os seguintes eixos principais não serão parados:

- Eixos de comando e
- eixos de posicionamento que se movimentam com POSA

O comando G4 do programa de peça é permitido na área Stop-Delay, ao passo que outros comandos do programa de peça que conduzem uma parada temporária (p. ex. WAITM) não são permitidos.

Como no caso de um movimento de percurso, o G4 ativa a área Stop-Delay e mantém sua ativação.

Exemplo: Intervenções do avanço

Se o override for reduzido em 6% antes da área Stop-Delay, então o override estará ativo na área Stop-Delay.

Se o override for reduzido de 100% para 6% na área Stop-Delay, então a área Stop-Delay é percorrida com 100% até o final e em seguida continuará com 6%.

O bloqueio de avanço não tem efeito na área Stop-Delay, apenas depois de sair da área Stop-Delay que é realizada a parada.

Sobreposição/aninhamento:

Se duas áreas Stop-Delay se sobreporem, uma a partir dos comandos de linguagem e outra do dado de máquina MD 11550: STOP_MODE_MASK, então se forma a maior área Stop-Delay possível.

Os itens a seguir regularizam a ação conjunta dos comandos de linguagem DELAYFSTON e DELAYFSTOF com aninhamentos e o fim da subrotina:

1. Com o fim da subrotina onde foi chamado o DELAYFSTON, se ativa implicitamente o DELAYFSTOF.
2. O DELAYFSTON da área Stop-Delay permanece sem efeito.
3. Se a subrotina1 chama a subrotina2 em uma área Stop-Delay, então a subrotina2 é uma área Stop-Delay completa. Especialmente o DELAYFSTOF está sem efeito na subrotina2.

Indicação

O REPOSA é um fim de subrotina e o DELAYFSTON é desselecionado em todo caso.

Se ocorrer um evento "brusco" de parada na "área Stop-Delay", então é desselecionada a "área Stop-Delay" inteira! Isto significa que se neste segmento do programa ocorrer uma outra parada qualquer, então esta será imediatamente executada. Somente com uma reprogramação (novo DELAYFSTON) é que pode ser iniciada uma nova área Stop-Delay.

Se a tecla Stop for pressionada antes da área Stop-Delay e a NCK deve ser executado para frenagem na área Stop-Delay, então o NCK pára na área Stop-Delay e a área Stop-Delay permanece desselecionada!

Quando se alcança uma área Stop-Delay com Override 0%, então a área Stop-Delay **não** será aceita!

Isto serve para todos eventos "suaves" de parada.

Com o STOPALL se pode frear na área Stop-Delay. Entretanto, com um STOPALL são ativados imediatamente todos os eventos de parada que até então eram prorrogados.

Variáveis de sistema

Uma área Stop-Delay pode ser detectada no programa de peça com $P_DELAYFST. Se o Bit 0 das variáveis de sistema estiver definido como 1, então neste momento a execução do programa de peça encontra-se em uma área Stop-Delay.

Uma área Stop-Delay pode ser detectada em ações sincronizadas com $AC_DELAYFST. Se o Bit 0 das variáveis de sistema estiver definido como 1, então neste momento a execução do programa de peça encontra-se em uma área Stop-Delay.

Compatibilidade

A definição prévia do dado de máquina MD 11550: STOP_MODE_MASK Bit 0 = 0 aplica uma área Stop-Delay implícita durante o grupo G331/G332 e quando for programado um movimento de percurso ou G4.

O Bit 0 = 1 permite a parada durante o grupo G331/G332 e quando for programado um movimento de percurso ou G4 (comportamento até o SW 6). Para a definição de uma área Stop-Delay devem ser utilizados os comandos DELAYFSTON/DELAYFSTOF.

8.5 Evitar posição de programa para SERUPRO (IPTRLOCK, IPTRUNLOCK)

Função

Para determinadas situações mecanicamente complicadas da máquina se faz necessário evitar a localização de blocos SERUPRO.

Com um indicador de interrupção programável existe a possibilidade de intervenção, na "Localização na posição de interrupção", antes da posição que não deve ser pesquisada.

Também se pode definir áreas do programa de peça que não devem ser pesquisadas, onde o NCK não deve entrar novamente. Com o cancelamento do programa o NCK marca o último bloco processado que pode ser localizado através da interface do operador HMI.

Sintaxe

```
IPTRLOCK
IPTRUNLOCK
```

Os comandos estão isolados em uma linha do programa de peça e permitem um indicador de interrupção programável

Significado

`IPTRLOCK`	Início do segmento de programa que não deve ser pesquisado
`IPTRUNLOCK`	Fim do segmento de programa que não deve ser pesquisado

Os dois comandos somente estão em programas de peça, mas **não** são permitidos em ações sincronizadas.

Exemplo

Aninhamento de segmentos de programa que não devem ser pesquisados em dois níveis de programa com IPTRUNLOCK implícito. O IPTRUNLOCK implícito na subrotina 1 finaliza a área que não deve ser pesquisada.

```
Código de programa                Comentário
N10010 IPTRLOCK()
N10020 R1 = R1 + 1
N10030 G4 F1                      ; Bloco de parada que inicia o segmento de programa
                                    sem localização
...
N10040 Subrotina2
...                               ; Interpretação da subrotina 2.
N20010 IPTRLOCK ()                ; Inativo, início repetido.
...
N20020 IPTRUNLOCK ()              ; Inativo, fim em outro nível.
N20030 RET
...
N10060 R2 = R2 + 2
N10070 RET                        ; Fim do segmento de programa que não deve ser
                                    pesquisado.
N100 G4 F2                        ; O programa principal é continuado.
```

Uma interrupção em 100 retorna novamente o indicador de interrupção.

Detectar e localizar áreas que não devem ser pesquisadas

Os segmentos de programa que não devem ser pesquisados são marcados com os comandos de linguagem IPTRLOCK e IPTRUNLOCK .

O comando IPTRLOCK congela o indicador de interrupção em um processamento principal de um bloco a bloco (SBL1). A partir daqui este bloco será nomeado como bloco de parada. Se ocorrer um cancelamento do programa após o IPTRLOCK, então este assim chamado bloco de parada pode ser localizado na interface do operador HMI.

Continuar novamente do bloco atual

O indicador de interrupção é posicionado no atual bloco com IPTRUNLOCK como o ponto de interrupção para o segmento de programa seguinte.

Após um destino de busca ser encontrado se pode realizar uma nova busca com o mesmo bloco de parada.

Um indicador de interrupção editado pelo usuário deve ser removido novamente através da HMI.

Regras de aninhamento

Os itens a seguir regularizam a ação conjunta dos comandos de linguagem `IPTRLOCK` e `IPTRUNLOCK` com aninhamentos e o fim da subrotina:

1. Com o fim da subrotina onde foi chamado o `IPTRLOCK`, se ativa implicitamente o `IPTRUNLOCK`.
2. O `IPTRLOCK` em uma área sem localização permanece sem efeito.
3. Se a subrotina1 chama a subrotina2 em uma área que não deve ser pesquisada, então a subrotina2 inteira não será pesquisada. Especialmente o `IPTRUNLOCK` está sem efeito na subrotina2.

Para mais informações sobre este assunto, veja o
/FB1/ Manual de funções básicas; BAG, Canal, Modo de operação do programa (K1).

Variável de sistema

Uma área que não deve ser pesquisada pode ser detectada no programa de peça com `$P_IPTRLOCK`.

Indicador de interrupção automático

A função de indicador de interrupção automático define automaticamente um tipo de acoplamento definido anteriormente para localização. Através de dado de máquina, para

- Transmissão eletrônica com `EGON`
- Acoplamento axial do valor mestre com `LEADON`

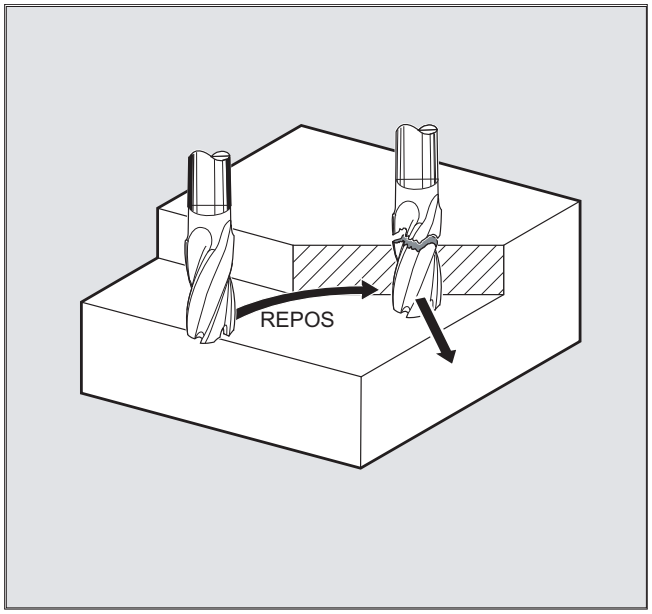
se ativa o indicador de interrupção automático. Se o indicador de interrupção programado e o ativado automaticamente através de dado de máquina se sobreporem, então se forma a maior área possível que não deve ser pesquisada.

8.6 Reaproximação no contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN)

Função

Se o programa em execução for interrompido durante a usinagem e a ferramenta estiver sendo recuada (p. ex. por conta de uma quebra de ferramenta ou para uma eventual medição), o contorno pode ser reaproximado em um ponto selecionável de forma controlada.

O comando REPOS atua como um retorno de subrotina (p. ex. via M17). Os blocos após a rotina de interrupção não serão mais executados.



Para interrupção da execução do programa, veja também a seção "Programação NC flexível" no capítulo "Rotina de interrupção" neste manual de programação.

Sintaxe

```
REPOSA RMI DISPR=…
REPOSA RMB
REPOSA RME
REPOSA RMN
REPOSL RMI DISPR=…
REPOSL RMB
REPOSL RME
REPOSL RMN
REPOSQ RMI DISPR=… DISR=…
REPOSQ RMB DISR=…
REPOSQ RME DISR=…
REPOSQA DISR=…
REPOSH RMI DISPR=… DISR=…
REPOSH RMB DISR=…
REPOSH RME DISR=…
REPOSHA DISR=…
```

Significado

Trajetória de aproximação

`REPOSA`	Aproximação em uma linha reta com todos eixos
`REPOSL`	Aproximação em uma linha reta
`REPOSQ DISR=…`	Aproximação em um quadrante com raio DISR
`REPOSQA DISR=…`	Aproximação com todos eixos em um quadrante com raio DISR
`REPOSH DISR=…`	Aproximação em um semicírculo de diâmetro DISR
`REPOSHA DISR=…`	Aproximação com todos eixos em um semicírculo de raio DISR

Ponto de reaproximação

`RMI`	Aproximar ponto de interrupção
`RMI DISPR=…`	Ponto de entrada com distância DISPR em mm/polegadas antes do ponto de interrupção
`RMB`	Aproximação do ponto inicial do bloco
`RME`	Aproximação do ponto final do bloco
`RME DISPR=…`	Aproximação do ponto final do bloco com distância DISPR antes do ponto final
`RMN`	Aproximação do ponto de percurso mais próximo
`A0 B0 C0`	Eixos em que se deve realizar a aproximação

Exemplo: Aproximação em uma linha reta, REPOSA, REPOSL

A ferramenta aproxima-se do ponto de reaproximação diretamente em uma linha reta.

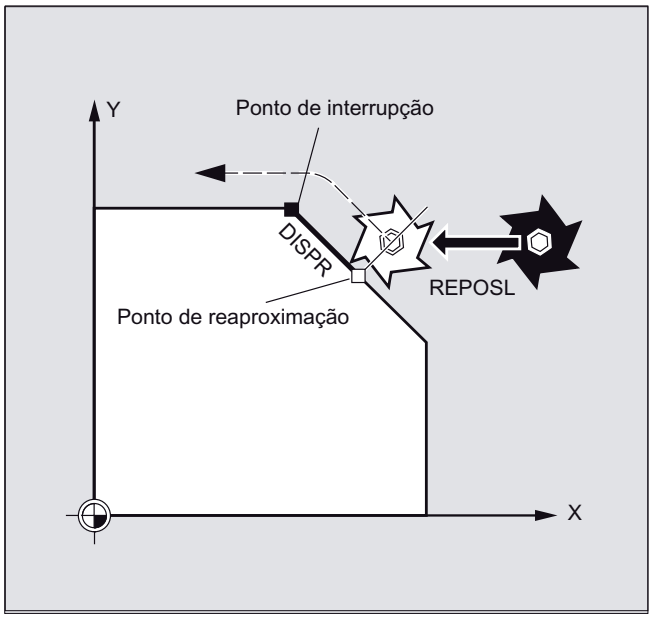
Com o REPOSA todos os eixos são movimentados automaticamente. Com o REPOSL podemos especificar os eixos que devem ser movimentados.

Exemplo:

```
REPOSL RMI DISPR=6 F400
```

ou

```
REPOSA RMI DISPR=6 F400
```

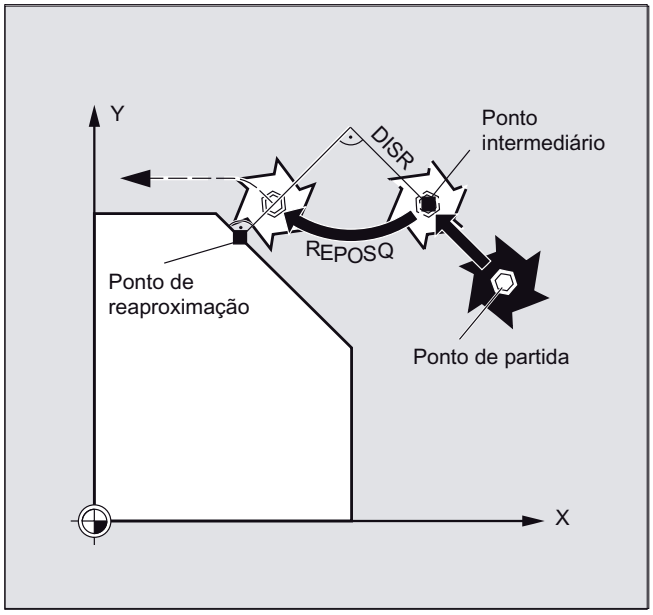


Exemplo: Aproximação em quadrante, REPOSQ, REPOSQA

A ferramenta aproxima-se do ponto de reaproximação em um quadrante com o raio `DISR=...` . O comando calcula automaticamente o ponto intermediário necessário entre o ponto de partida e o ponto de reaproximação.

Exemplo:

```
REPOSQ RMI DISR=10 F400
```

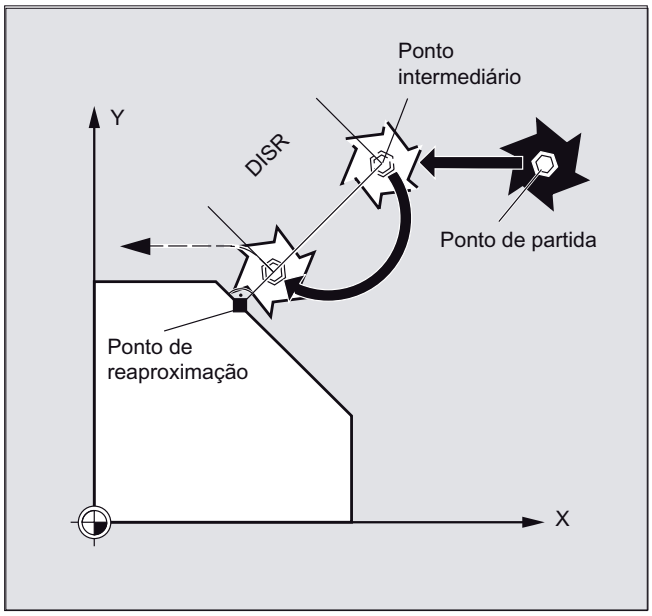


Exemplo: Aproximação em semicírculo, REPOSH, REPOSHA

A ferramenta aproxima-se do ponto de reaproximação em um semicírculo com o diâmetro `DISR=...` . O comando calcula automaticamente o ponto intermediário necessário entre o ponto de partida e o ponto de reaproximação.

Exemplo:

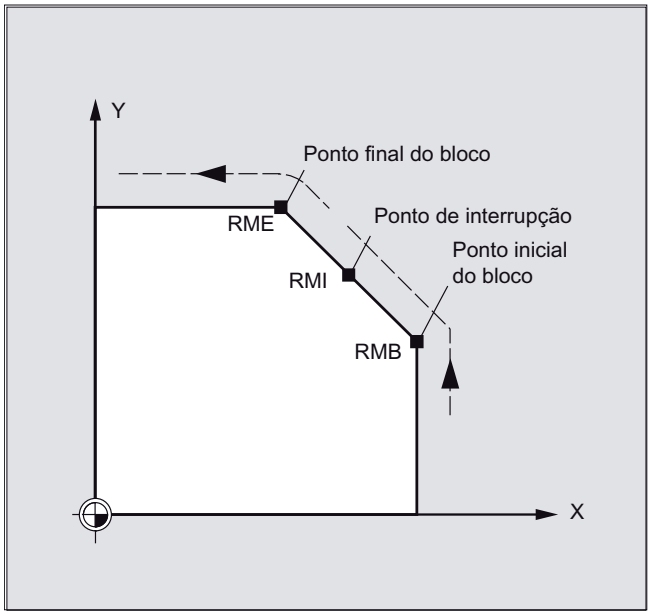
```
REPOSH RMI DISR=20 F400
```



Definir o ponto de reaproximação (não para aproximação SERUPRO com RMN)

Dependendo do bloco NC onde o processo do programa foi interrompido, pode-se escolher entre um dos três pontos de reaproximação:

- RMI, ponto de interrupção
- RMB, ponto inicial do bloco ou o último ponto final
- RME, ponto final do bloco



Com `RMI DISPR=...` ou com `RME DISPR=...` podemos definir o ponto de reaproximação que está antes do ponto de interrupção ou antes do ponto final do bloco.

Com `DISPR=...` descrevemos a trajetória do contorno em mm/polegadas em torno do qual está o ponto de reaproximação antes do ponto de interrupção ou o ponto final. Este ponto no máximo pode estar no ponto inicial do bloco, também para valores mais altos.

Se nenhum `DISPR=...` for programado, aplica-se `DISPR=0` e com isso programa-se o ponto de interrupção (para `RMI`) ou o ponto final do bloco (para `RME`).

Sinal do DISPR

O sinal do DISPR é avaliado. Se o sinal for positivo permanece o procedimento descrito anteriormente.

Se o sinal for negativo a reaproximação será realizada após o ponto de interrupção e para RMB atrás do ponto de partida.

A distância do ponto de interrupção para o ponto de aproximação resulta do valor do DISPR. Para valores maiores, este ponto também pode estar no máximo no ponto final do bloco.

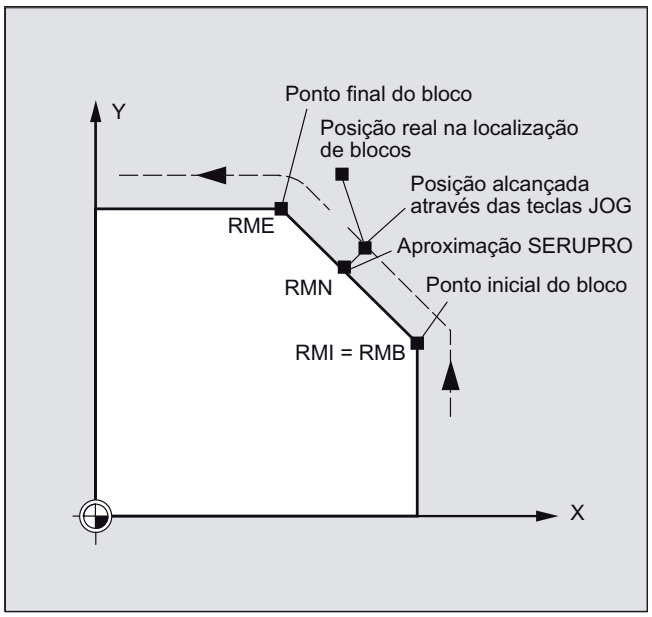
Exemplo de aplicação:

Através de um sensor se detecta a aproximação de uma garra de fixação. É iniciada uma ASUP com a qual se desvia da garra de fixação.

Em seguida é feito o reposicionamento em um ponto atrás da garra de fixação com DISPR negativo e o programa é continuado.

Aproximação SERUPRO com RMN

Se para a usinagem em uma posição qualquer for forçado um cancelamento, então será realizada a aproximação SERUPRO sob RMN pelo curso mais curto até o ponto de interrupção, depois será executado apenas o curso restante. Para isso o usuário inicia um processo SERUPRO no bloco de interrupção e realiza o posicionamento através das teclas JOG antes do ponto danificado do bloco de destino.



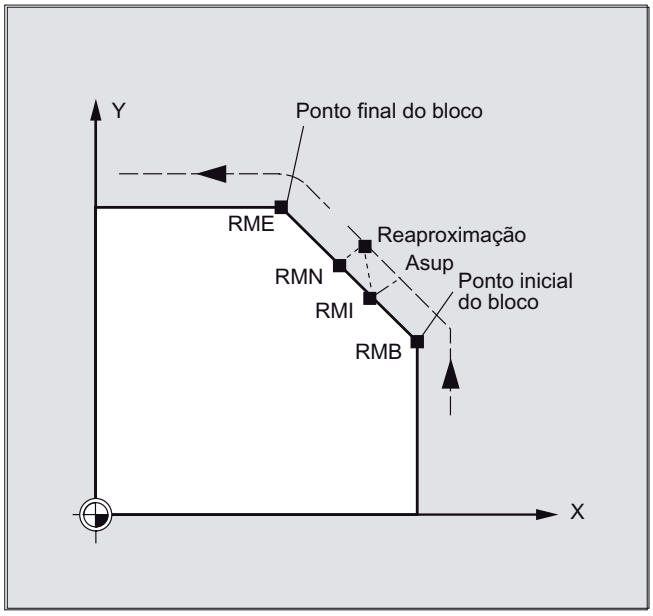
Indicação

SERUPRO

Para o SERUPRO o RMI e o RMB são idênticos.O RMN não é restrito apenas ao SERUPRO, mas normalmente aplicado.

Aproximação do ponto de percurso mais próximo RMN

Quando o REPOSA é interpretado, após uma interrupção, o bloco de reaproximação com RMN não é completamente reiniciado, apenas executa-se o curso restante. É realizada a aproximação do ponto de percurso mais próximo do bloco interrompido.



Estado do modo REPOS válido

O modo REPOS válido do bloco interrompido pode ser lido através de ações sincronizadas com a variável $AC_REPOS_PATH_MODE:

0: Aproximação não definida

1 RMB: Aproximação no início

2 RMI: Aproximação no ponto de interrupção

3 RME: Aproximação no ponto final do bloco

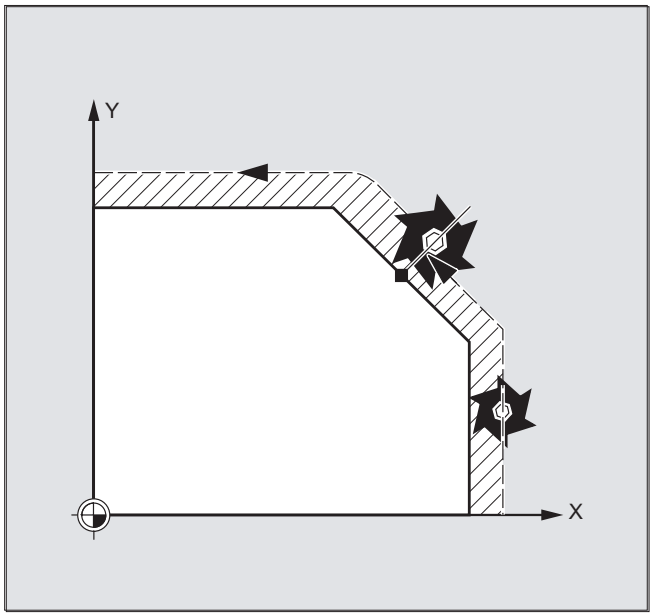
4 RMN: Aproximação no ponto de percurso mais próximo do bloco interrompido.

Aproximação com nova ferramenta

Se interrompemos a execução do programa devido uma quebra de ferramenta:

Com a programação do novo número D, o programa será continuado a partir do ponto de reaproximação com os valores alterados da correção da ferramenta.

No caso de valores alterados da correção da ferramenta, o ponto de interrupção possivelmente não poderá mais ser aproximado. Neste caso é aproximado o ponto do novo contorno mais próximo do ponto de interrupção (eventualmente modificado pelo DISPR).



Aproximar contorno

O movimento com que a ferramenta é reaproximada no contorno pode ser programado. Os endereços dos eixos de movimento indicam o valor zero.

Com os comandos `REPOSA`, `REPOSQA` e `REPOSHA` todos eixos são reposicionados automaticamente. Não será necessário especificar nenhum eixo.

Com a programação do `REPOSL`, `REPOSQ` e `REPOSH` todos os eixos geométricos são aproximados automaticamente, portanto, também sem a especificação no comando. Todos demais eixos devem ser especificados no comando.

Para os movimentos circulares REPOSH e REPOSQ aplica-se:

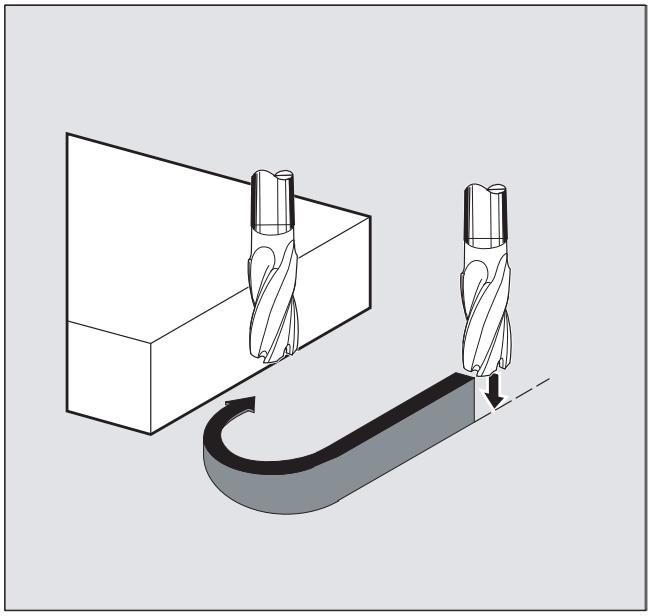
O círculo é percorrido com o plano de trabalho `G17` até `G19` especificado.

Se for especificado o terceiro eixo geométrico (sentido da penetração) no bloco de aproximação, então para este caso o ponto de reaproximação é aproximado em uma linha espiral, onde a posição da ferramenta e a posição programada no sentido de avanço não coincidem.

Nos seguintes casos automaticamente se

passa para a aproximação linear `REPOSL`:

- Se não foi especificado nenhum valor para `DISR`.
- Não existe nenhum sentido de aproximação definido (interrupção de programa em um bloco sem informação de movimento).
- Para sentido de aproximação perpendicular ao atual plano de trabalho.



8.7 Controle na condução do movimento

8.7.1 Correção percentual de solavancos (JERKLIM)

Função

Com o comando NC `JERKLIM` é possível que o solavanco máximo permitido de um eixo ajustado via dado de máquina seja reduzido ou acelerado no movimento de percurso em seções de programa críticas.

Pré-requisito

O modo de aceleração SOFT deverá estar ativo.

Efeito

A função tem efeito:

- em modos de operação AUTOMÁTICO.
- somente em eixos de percurso.

Sintaxe

```
JERKLIM[<eixo>]=<valor>
```

Significado

`JERKLIM`:	Comando para correção de solavancos
`<eixo>`:	Eixo da máquina cujo valor limite de solavanco deve ser adaptado.
`<valor>`:	Valor de correção percentual, relacionado ao solavanco de eixo máximo configurado no movimento de percurso (MD32431 $MA_MAX_AX_JERK). Faixa de valores: 1 ... 200 O valor 100 não tem nenhuma influência sobre o solavanco.

Indicação

O comportamento do JERKLIM no fim do programa de peça e no Reset de canal é configurado com Bit 0 no dado de máquina MD32320 $MA_DYN_LIMIT_RESET_MASK:

- Bit 0 = 0:
  O valor programado para o JERKLIM é resetado em 100 % com o Reset de canal/M30.
- Bit 0 = 1:
  O valor programado para JERKLIM é preservado além do Reset de canal/M30.

Exemplo

```
Código de programa          Comentário
...
N60 JERKLIM[X]=75           ; O carro de eixo no sentido X deve ser
                              acelerado/desacelerado apenas com o máximo de 75% de
                              solavancos permitido para o eixo.
...
```

8.7.2 Correção percentual da velocidade (VELOLIM)

Função

Com o comando NC `VELOLIM` é possível reduzir a velocidade máxima possível de um eixo/fuso em modo de eixo e reduzir a rotação máxima possível dependente da gama de velocidades de um fuso em modo de fuso (modo de controle de rotação M3, M4, M5 e modo de posicionamento SPOS, SPOSA, M19) em seções de programa críticas, por exemplo, para reduzir o esforço da máquina ou para melhorar a qualidade de usinagem.

Efeito

A função tem efeito:

- em modos de operação AUTOMÁTICO.
- em eixos de percurso e eixos de posicionamento.
- em fusos em modo de fuso/eixo

Sintaxe

```
VELOLIM[<eixo/fuso>]=<valor>
```

Significado

VELOLIM:	Comando para correção da velocidade
<eixo/fuso>:	Eixo de máquina ou fuso, cujo valor limite de velocidade ou de rotação deve ser adaptado. **VELOLIM para fusos** Através do dado de máquina (MD30455 $MA_MISC_FUNCTION_MASK, Bit 6) é possível definir no programa de peça se a programação do VELOLIM deve atuar independentemente do atual uso como fuso ou eixo (Bit 6 = 1) ou se ele pode ser programado separadamente para cada modo de operação (Bit 6 = 0). Se for projetado um efeito separado, então a seleção é realizada na programação através do identificador: • Identificador de fuso S<n> para modos de operação de fuso • Identificador de eixo, p. ex. "C", para o modo de eixo
<valor>:	Valor de correção percentual O valor de correção refere-se: • em eixos / fusos ao modo de eixo (se MD30455 Bit 6 = 0): na velocidade de eixo máxima projetada (MD32000 $MA_MAX_AX_VELO). • em eixos ao modo de fuso ou de eixo (se MD30455 Bit 6 = 1): na rotação máxima da gama de velocidade ativa (MD35130 $MA_GEAR_STEP_MAX_VELO_LIMIT[<n>]) Faixa de valores: 1 ... 100 O valor 100 não tem nenhuma influência sobre a velocidade e rotação.

Indicação

Comportamento no fim do programa de peça e Reset de canal

O comportamento do VELOLIM no fim do programa de peça e no Reset de canal é configurado com Bit 0 no dado de máquina MD32320 $MA_DYN_LIMIT_RESET_MASK:

- Bit 0 = 0:
  O valor programado para o VELOLIM é resetado em 100 % com o Reset de canal/M30.
- Bit 0 = 1:
  O valor programado para VELOLIM é preservado além do Reset de canal/M30.

Indicação

VELOLIM para fusos em ações síncronas

Na programação do VELOLIM em ações síncronas é feita a diferenciação entre modo de fuso e modo de eixo. Independente do identificador utilizado na programação, a rotação no modo de fuso e a velocidade no modo de eixo são limitadas de maneira uniforme.

Diagnóstico

Diagnóstico do VELOLIM em modo de fuso

Um limite de rotação ativo através do VELOLIM (menor que 100 %) pode ser detectado em modo de fuso através da leitura das variáveis de sistema $AC_SMAXVELO e $AC_SMAXVELO_INFO.

No caso de um limite o $AC_SMAXVELO retorna o limite de rotação gerado pelo VELOLIM. A variável $AC_SMAXVELO_INFO, neste caso, retorna o valor "16" como detecção para a causa de limite VELOLIM.

Exemplos

Exemplo 1: Limitação de velocidade de eixos de máquina

```
Código de programa          Comentário
...
N70 VELOLIM[X]=80           ; O carro de eixo no sentido X somente deve ser
                              movimentado com o máximo de 80% de velocidade
                              permitida para o eixo.
...
```

Exemplo 2: Limite de rotação do fuso

```
Código de programa          Comentário
N05 VELOLIM[S1]=90          ; Limitação da rotação máxima do fuso 1 em 90% de
                              1000 rpm.
...
N50 VELOLIM[C]=45           ; Limitação da rotação em 45% de 1000 rpm, C é o
                              identificador de eixo do S1.
...
```

Dados de projeção para o fuso 1 (AX5):

MD35130 $MA_GEAR_STEP_MAX_VELO_LIMIT[1,AX5]=1000	; Rotação máxima da gama de velocidade 1 = 1000 rpm
MD30455 $MA_MISC_FUNCTION_MASK[AX5] = 64	; Bit 6 = 1: A programação do VELOLIM tem efeito geral para modo de fuso e modo de eixo, independentemente do identificador programado.

8.7.3 Exemplo de programa para JERKLIM e VELOLIM

O programa a seguir apresenta um exemplo de aplicação para a limitação em porcentagem do número de solavancos e da velocidade:

```
Código de programa                    Comentário
N1000 G0 X0 Y0 F10000 SOFT G64
N1100 G1 X20 RNDM=5 ACC[X]=20
ACC[Y]=30
N1200 G1 Y20 VELOLIM[X]=5             ; O carro de eixo no sentido X somente deve
                                        ser movimentado com o máx. 5% da
                                        velocidade permitida para o eixo.
JERKLIM[Y]=200                        ; O carro de eixo no sentido Y deve ser
                                        acelerado/desacelerado no máx. com 200% de
                                        solavancos permitido para o eixo.
N1300 G1 X0 JERKLIM[X]=2              ; O carro de eixo no sentido X deve ser
                                        acelerado/desacelerado apenas como máx. 2%
                                        de solavancos permitido para o eixo.
N1400 G1 Y0
M30
```

8.8 Tolerância de contorno/orientação programável (CTOL, OTOL, ATOL)

Função

Com os comandos CTOL, OTOL e ATOL as tolerâncias de usinagem definidas através de dados de máquina e de ajuste podem ser adaptadas no programa NC para as funções de compressor (COMPON, COMPCURV, COMPCAD), os tipos de suavização G642, G643, G645, OST e a suavização de orientação ORISON.

Os valores programados permanecem válidos até serem reprogramados ou cancelados através da atribuição de um valor negativo. Mais adiante, eles são cancelados no fim do programa, Reset de canal, BAG-Reset, NCK-Reset (reinicialização) e Power On (partida a frio). Depois do cancelamento, são aplicados novamente os valores dos dados de máquina e de ajuste.

Sintaxe

```
CTOL=<valor>
OTOL=<valor>
ATOL[<eixo>]=<valor>
```

Significado

CTOL Comando para programação da **tolerância de contorno**

O CTOL é válido para:

- todas funções de compressor
- todos tipos de suavização exceto G641 e G644

<valor>: O valor para a tolerância de contorno é uma especificação de comprimento.

Tipo: REAL

Unidade: Inch/mm (em função do atual ajuste da indicação de dimensões)

OTOL Comando para programação da **tolerância de orientação**

O OTOL é válido para:

- todas funções de compressor
- Suavização de orientação ORISON
- todos tipos de suavização exceto G641, G644 e OSD

<valor>: O valor para a tolerância de orientação é uma especificação angular.

Tipo: REAL

Unidade: Graus

ATOL Comando para programação de uma **tolerância específica de eixo**

O ATOL é válido para:

- todas funções de compressor
- Suavização de orientação ORISON
- todos tipos de suavização exceto G641, G644 e OSD

<eixo>: Nome do eixo, para o qual deve ser programada uma tolerância de eixo

<valor>: O valor para a tolerância de eixo pode ser uma especificação de comprimento ou angular dependendo do tipo de eixo (eixo linear ou rotativo).

Tipo:	REAL	
Unidade:	para eixos lineares:	Inch/mm (em função do atual ajuste da indicação de dimensões)
	para eixos rotativos:	Graus

Indicação

O CTOL e o OTOL têm prioridade sobre o ATOL.

Condições gerais

Frames escalonáveis

Os Frames escalonáveis atuam sobre as tolerâncias programadas da mesma forma como nas posições de eixo, isto é, a tolerância relativa permanece a mesma.

Exemplo

```
Código de programa                  Comentário
COMPCAD G645 G1 F10000              ; Ativa a função de compressor COMPCAD.
X... Y... Z...                      ; Aqui atuam os dados de máquina e de ajuste.
X... Y... Z...
X... Y... Z...
CTOL=0.02                           ; A partir deste ponto atua a tolerância de
                                      contorno de 0,02 mm.
X... Y... Z...
X... Y... Z...
X... Y... Z...
ASCALE X0.25 Y0.25 Z0.25            ; A partir deste ponto atua a tolerância de
                                      contorno de 0,005 mm.
X... Y... Z...
X... Y... Z...
X... Y... Z...
```

Código de programa	Comentário
CTOL=-1	; A partir destes ponto atuam novamente os dados de máquina e de ajuste.
X... Y... Z...	
X... Y... Z...	
X... Y... Z...	

Outras informações

Leitura de valores de tolerância

Para aplicações avançadas ou para diagnóstico, independentemente do tipo de estado, é possível realizar a leitura através de variáveis de sistema das atuais tolerâncias aplicadas para as funções de compressor (COMPON, COMPCURV, COMPCAD), os tipos de suavização G642, G643, G645, OST e a suavização de orientação ORISON.

- Em ações síncronas ou com parada de pré-processamento no programa de peça através das variáveis de sistema:

$AC_CTOL	Tolerância de contorno, que estava ativa durante a preparação do atual bloco de processamento principal Se nenhuma tolerância de contorno estava ativa, o `$AC_CTOL` retorna a raiz da soma dos quadrados das tolerâncias dos eixos geométricos.
$AC_OTOL	Tolerância de orientação, que estava ativa durante a preparação do atual bloco de processamento principal Se nenhuma tolerância de orientação estava ativa, durante uma transformação de orientação ativa, o `$AC_OTOL` retorna a raiz da soma dos quadrados das tolerâncias dos eixos de orientação, senão retorna o valor "-1".
$AA_ATOL[<eixo>]	Tolerância de eixo, que estava ativa durante a preparação do atual bloco de processamento principal Se uma tolerância de contorno estiver ativa, o `$AA_ATOL[<eixo geométrico>]` retorna a tolerância de contorno dividida pela raiz do número de eixos geométricos. Se uma tolerância de orientação e uma transformação de orientação estiverem ativas, o `$AA_ATOL[<eixo de orientação>]` retorna a tolerância de orientação dividida pela raiz do número de eixos de orientação.

Indicação

Se nenhum valor de tolerância foi programado, então as variáveis $A não estão o suficientemente diferenciadas para permitir a distinção de possíveis diferenças de tolerância nas diversas funções, pois elas apenas podem apontar um valor.

Tais casos podem ocorrer quando os dados de máquina e de ajuste ajustam diferentes tolerâncias para funções de compressor, suavização e suavização de orientação. Neste caso as variáveis retornam o maior valor, que ocorre nas funções ativas no momento.

Por exemplo, se uma função de compressor estiver ativa com a tolerância de orientação 0,1° e uma suavização de orientação ORISON estiver ativa com 1°, a variável $AC_OTOL retorna o valor "1". Quando a suavização de orientação for desativada, apenas lemos o valor "0,1".

- Sem parada de pré-processamento no programa de peça através das variáveis de sistema:

$P_CTOL	Tolerância de contorno programada
$P_OTOL	Tolerância de orientação programada
$PA_ATOL	Tolerância de eixo programada

Indicação

Se nenhum valor de tolerância for programado, então as variáveis $P retornam o valor "-1".

8.9 Tolerância em movimentos G0 (STOLF)

Fator de tolerância G0

Os movimentos G0 (avanço rápido, movimentos de penetração) podem ser deslocados com maior tolerância se comparados com a usinagem da peça de trabalho. Isto tem a vantagem de encurtar os tempos de deslocamento para os movimentos G0.

O ajuste das tolerâncias em movimentos G0 é realizada através da projeção do fator de tolerância do G0 (MD20560 $MC_G0_TOLERANCE_FACTOR).

O fator de tolerância do G0 somente tem efeito se:

- uma das seguintes funções estiver ativa:
  - Funções de compressor: COMPON, COMPCURV e COMPCAD
  - Funções de suavização: G642 e G645
  - Suavização de orientação: OST
  - Suavização de orientação: ORISON
  - Suavização na orientação relativa à trajetória: ORIPATH
- vários blocos G0 consecutivos (≥ 2).

  Para um bloco G0 individual o fator de tolerância não tem nenhum efeito, pois **na transição** de um movimento não G0 para um movimento G0 (e inversamente) sempre age a "**menor tolerância**" (tolerância da usinagem da peça de trabalho)!

Função

Através da programação do `STOLF` no programa de peça é possível sobrescrever temporariamente o fator de tolerância G0 (MD20560). O valor no MD20560 não é alterado com isso. Após o Reset e o fim do programa de peça o fator de tolerância projetado volta a ter efeito.

Sintaxe

```
STOLF=<fator de tolerância>
```

Significado

`STOLF`:	Comando para programação do fator de tolerância G0
`<fator de tolerância>`:	Fator de tolerância G0 O fator pode ser tanto maior que 1 como menor que 1. Normalmente podem ser ajustadas tolerâncias maiores para os movimentos G0. Com o `STOLF=1.0` (corresponde ao valor padrão projetado) as tolerâncias para os movimentos G0 são as mesmas ativas como os movimentos não G0.

Variáveis de sistema

O fator de tolerância do G0 ativo no programa de peça e no atual bloco IPO pode ser lido através de variáveis de sistema.

- Em ações síncronas ou com parada de pré-processamento no programa de peça através da variável de sistema:

$AC_STOLF	Fator de tolerância G0 ativo Fator de tolerância G0 que estava ativo durante a preparação do atual bloco de processamento principal.

- Sem parada de pré-processamento no programa de peça através da variável de sistema:

$P_STOLF	Fator de tolerância G0 programado

Se nenhum valor foi programado com `STOLF` no programa de peça, então estas duas variáveis de sistema retornam o valor ajustado através do MD20560 $MC_G0_TOLERANCE_FACTOR.

Se em um bloco não houver nenhum avanço rápido (G0) ativo, estas variáveis de sistema sempre retornam o valor 1.

Exemplo

```
Código de programa                  Comentário
COMPCAD G645 G1 F10000              ; Função de compressor COMPCAD
X... Y... Z...                      ; Aqui atuam os dados de máquina e de ajuste.
X... Y... Z...
X... Y... Z...
G0 X... Y... Z...
G0 X... Y... Z...                   ; Aqui atua o dado de máquina $MC_G0_TOLERANCE_FACTOR
                                      (p. ex. =3), e também uma tolerância de suavização de
                                      $MC_G0_TOLERANCE_FACTOR*$MA_COMPRESS_POS_TOL.
CTOL=0.02
STOLF=4
G1 X... Y... Z...                   ; A partir deste ponto atua a tolerância de contorno de
                                      0,02mm.
X... Y... Z...
X... Y... Z...
G0 X... Y... Z...
X... Y... Z...                      ; A partir daqui atua um fator de tolerância G0 de
                                      valor 4, isto é, uma tolerância de contorno de
                                      0,08mm.
```

Acoplamentos de eixo

9

9.1 Movimento acoplado (TRAILON, TRAILOF)

Função

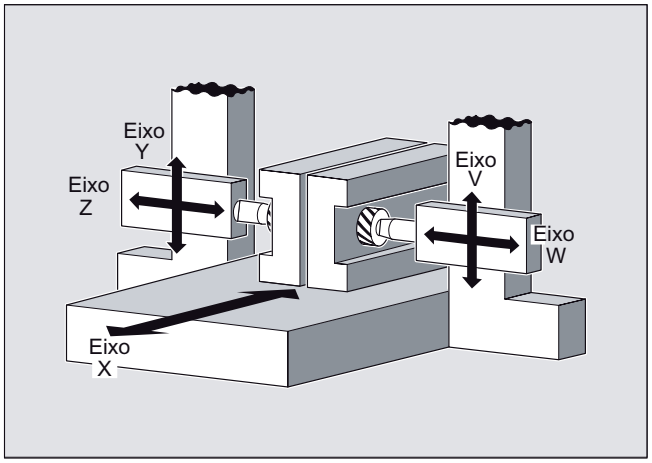
Ao ser movimentado um eixo mestre definido, os eixos acoplados (= eixos escravos) à ele associados são movimentados nos percursos descritos pelo eixo mestre, sob a consideração de um fator de acoplamento.

Eixo mestre e eixos escravos juntos formam um grupo de movimento acoplado.

Áreas de aplicação

- Movimento de um eixo através de um eixo simulado. O eixo mestre é um eixo simulado e o eixo acoplado um eixo real. Com isso o eixo real pode ser movimentado sob a consideração de um fator de acoplamento.
- Usinagem bilateral com 2 grupos de movimento acoplado:

  1. Eixo mestre Y, eixo acoplado V
  2. Eixo mestre Z, eixo acoplado W



Sintaxe

```
TRAILON(<eixo escravo>,<eixo mestre>,<fator de acoplamento>)
TRAILOF(<eixo escravo>,<eixo mestre>,<eixo escravo 2>)
TRAILOF(<eixo escravo>)
```

Significado

`TRAILON`	Comando para ativar e definir um grupo de movimento acoplado Efeito: modal
`<eixo escravo>`	Parâmetro 1: Denominação do eixo de movimento acoplado **Nota:** Um eixo de movimento acoplado também pode ser eixo mestre para outros eixos acoplados. Desse modo podem ser construídos diferentes grupos de movimento acoplado.
`<eixo mestre>`	Parâmetro 2: Denominação do eixo mestre
`<fator de acoplamento>`	Parâmetro 3: Fator de acoplamento O fator de acoplamento estabelece a relação desejada dos cursos do eixo acoplado e eixo mestre: <fator de acoplamento> = Curso do eixo acoplado/curso do eixo mestre Tipo: REAL Pré-ajuste: 1 A especificação de um valor negativo desencadeia um movimento inverso de percurso dos eixos mestres e acoplados. Se o fator de acoplamento não for especificado na programação, então será aplicado automaticamente o fator de acoplamento 1.

`TRAILOF`	Comando para desativar um grupo de movimento acoplado
	Efeito: modal
	O `TRAILOF` com 2 parâmetros apenas desativa o acoplamento para o eixo mestre indicado:
	`TRAILOF(<eixo escravo>,<eixo mestre>)`
	Quando um eixo acoplado possui 2 eixos mestres, para desativação dos dois acoplamentos o `TRAILOF` pode ser chamado com 3 parâmetros:
	`TRAILOF(<eixo escravo>,<eixo mestre>,<eixo escravo 2>)`
	O mesmo resultado é oferecido pela programação do `TRAILOF` sem a indicação de um eixo mestre:
	`TRAILOF(<eixo escravo>)`

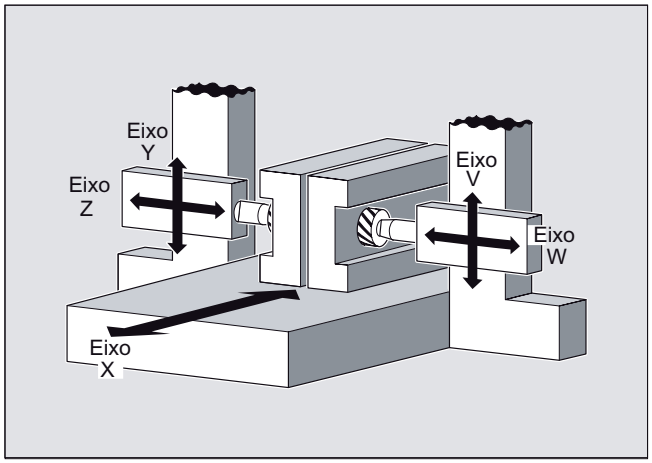
Indicação

O movimento acoplado sempre é realizado no sistema de coordenadas básico (BCS).

O número de grupos de movimento acoplado ativados simultaneamente apenas é limitado pelas opções de combinação dos eixos disponíveis na máquina.

Exemplo

A peça de trabalho deve ser usinada nos dois lados com o conjunto de eixos mostrado. Para isso crie 2 grupos de movimento acoplado.



Código de programa	Comentário
…	
N100 TRAILON(V,Y)	; Ativação do 1° grupo de movimento acoplado
N110 TRAILON(W,Z,-1)	; Ativação do 2° grupo de movimento acoplado. Fator negativo de acoplamento: O eixo acoplado movimenta-se no sentido oposto ao eixo mestre.
N120 G0 Z10	; Penetração do eixo Z e eixo W no sentido oposto do eixo.
N130 G0 Y20	; Penetração do eixo Y e eixo V no mesmo sentido do eixo.
…	
N200 G1 Y22 V25 F200	; Sobreposição de um movimento dependente e um independente do eixo acoplado V.
…	
TRAILOF(V,Y)	; Desativação do 1° grupo de movimento acoplado.
TRAILOF(W,Z)	; Desativação do 2° grupo de movimento acoplado.

Outras informações

Tipos de eixos

Um grupo movimento acoplado pode conter um número qualquer de combinações de eixos lineares e rotativos. Como eixo mestre também se pode definir um eixo simulado.

Eixos acoplados

Um eixo acoplado pode ser atribuído no máximo à 2 eixos mestres simultaneamente. A atribuição é realizada em diferentes grupos de movimento acoplado.

Um eixo acoplado pode ser programado com todos comandos de movimento disponíveis (G0, G1, G2, G3, …). Além do curso independente definido, o eixo acoplado percorre os cursos derivados de seus eixos mestres com os fatores de acoplamento.

Limitação de dinâmica

A limitação de dinâmica depende do tipo de ativação do grupo de movimento acoplado:

- Ativação no programa de peça

  Se a ativação é realizada no programa de peça e todos os eixos mestres são eixos de programa no canal ativado, a dinâmica de todos eixos acoplados será considerada durante o deslocamento dos eixos mestres, para que nenhum eixo acoplado seja sobrecarregado.

  Se a ativação é realizada no programa de peça com eixos mestres, que não estão ativos como eixos de programa no canal ativo ($AA_TYP ≠ 1), a dinâmica do eixo acoplado não será considerada durante o deslocamento dos eixos mestres. Com isso pode ocorrer uma sobrecarga nos eixos acoplados com uma dinâmica um pouco menor do que a necessária para o acoplamento.

- Ativação na ação síncrona

  Se a ativação é realizada em uma ação síncrona, a dinâmica dos eixos acoplados não será considerada durante o deslocamento dos eixos mestres. Com isso pode ocorrer uma sobrecarga nos eixos acoplados com uma dinâmica um pouco menor do que a necessária para o acoplamento.

⚠ CUIDADO

Quando um grupo de movimento acoplado

- em ações síncronas
- no programa de peça com eixos mestres, que não são eixos de programa no canal do eixo acoplado,

é ativado, então o usuário/fabricante da máquina tem uma responsabilidade especial para empregar medidas adequadas, para que não ocorra nenhuma sobrecarga dos eixos acoplados através dos movimentos do eixo mestre.

Estado do acoplamento

O estado de acoplamento de um eixo pode ser consultado no programa de peça com as variáveis de sistema:

$AA_COUP_ACT[<eixo>]

Valor	Significado
0	Nenhum acoplamento ativo
8	Movimento acoplado ativo

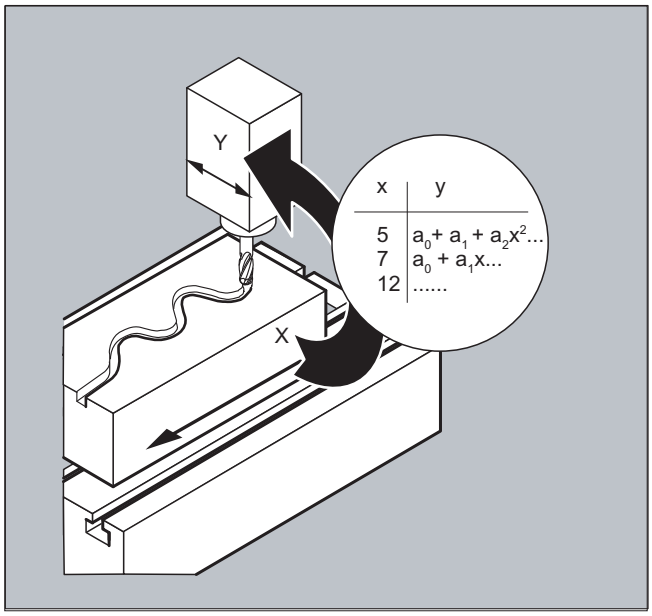
9.2 Tabelas de curvas (CTAB)

Função

Com a ajuda das tabelas de curvas é possível programar relações de posição e velocidade entre dois eixos (eixo mestre e eixo escravo). A definição da tabela de curvas é realizada no programa de peça.

Aplicação

As tabelas de curvas substituem os discos de cames. Neste caso a tabela de curvas forma a base para o acoplamento axial de valor mestre, no qual se estabelece a relação funcional entre valor mestre e valor escravo: Conforme a programação o comando calcula, a partir das posições atribuídas para eixo mestre e eixo escravo, um polinômio que corresponde ao disco de cames.

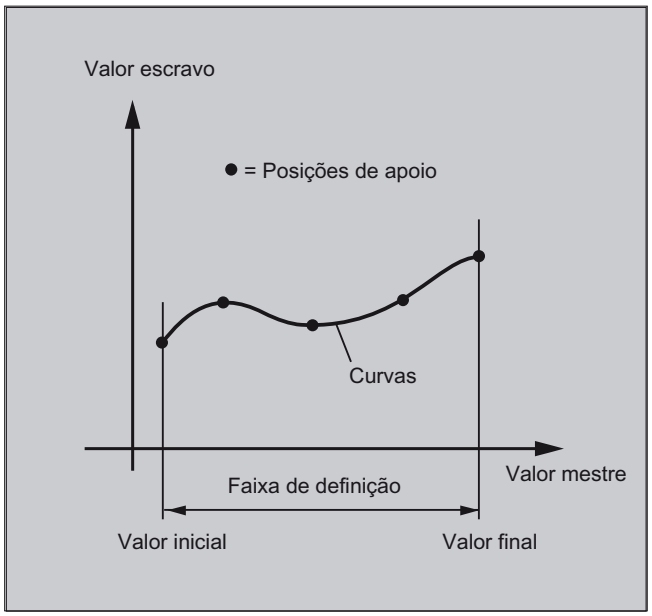


9.2.1 Definição de tabelas de curvas (CTABDEF, CATBEND)

Função

Uma tabela de curvas representa um programa de peça ou um segmento do programa de peça que está marcado no início com o comando `CTABDEF` e no fim com o `CTABEND`.

Neste segmento do programa de peça são associadas posições únicas de eixos escravos para cada posição do eixo mestre através de instruções de movimento, e estas posições servem de pontos de apoio para o cálculo da curva em forma de um polinômio de até 5º grau.



Pré-requisito

Para a definição de tabelas de curvas deve ser reservado o espaço de memória necessário através da configuração de dado de máquina adequada (→ Fabricante da máquina!).

Sintaxe

```
CTABDEF(<eixo escravo>,<eixo mestre>,<n>,<periodicidade>[,<local de
armazenamento>])
...
CTABEND
```

Significado

`CTABDEF ( )`	Início da definição de tabela de curvas
`CTABEND`	Fim da definição de tabela de curvas
`<eixo escravo>`	Eixo, cujo movimento deve ser calculado através da tabela de curvas
`<eixo mestre>`	Eixo, que fornece os valores mestres para o cálculo do movimento do eixo escravo
`<n>`	Número (**ID**) da tabela de curvas O número de uma tabela de curvas é único e independe do local de armazenamento. Não pode existir nenhuma tabela de mesmo número na memória estática e na memória dinâmica do NC.
`<periodicidade>`	Periodicidade da tabela `0` A tabela não é periódica (é processada apenas uma vez, mesmo para eixos rotativos) `1` A tabela é periódica relativa ao eixo mestre `2` A tabela é periódica relativa ao eixo mestre e ao eixo escravo
`<local de armazenamento>`	Especificação do local de armazenamento (opcional) `"SRAM"` A tabela de curvas é criada na memória **estática** do NC. `"DRAM"` A tabela de curvas é criada na memória **dinâmica** do NC. **Nota:** Se não for programado nenhum valor para este parâmetro, então é utilizado o local de armazenamento padrão definido no dado MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE.

Indicação

Sobrescrever

Uma tabela de curvas é sobregravada assim que uma nova definição de tabela utilizar seu número (`<n>`) (Exceção: a tabela de curvas está ativa em um acoplamento de eixo ou é bloqueada com `CTABLOCK`). **Na sobregravação não é emitido nenhum aviso correspondente!**

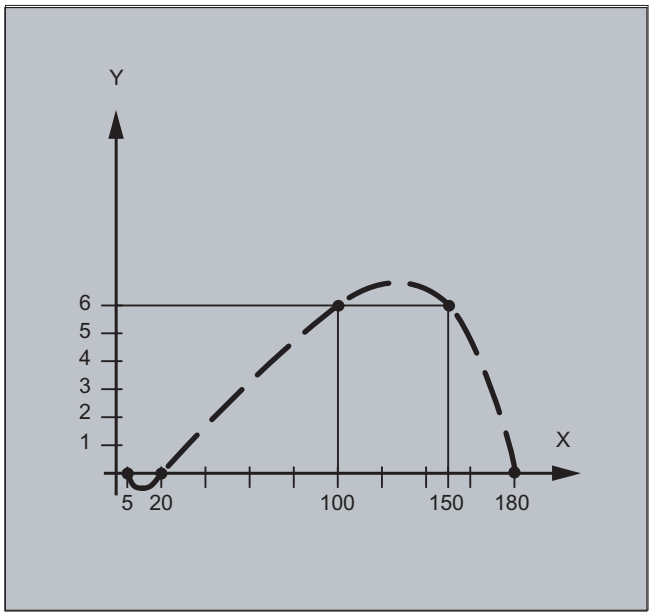
Exemplos

Exemplo 1: Segmento de programa como definição de tabela de curvas

Um segmento do programa deve ser utilizado sem alterações para definição de uma tabela de curvas. O comando ali contido e responsável pela parada do pré-processamento `STOPRE` pode permanecer e ele será imediatamente ativado quando o segmento do programa não for utilizado para definição da tabela e se `CTABDEF` e `CTABEND` foram removidos.

Código de programa	Comentário
…	
CTABDEF(Y,X,1,1)	; Definição de uma tabela de curvas.
…	
IF NOT ($P_CTABDEF)	
STOPRE	
ENDIF	
…	
CTABEND	

Exemplo 2: Definição de uma tabela de curvas não periódica



Código de programa	Comentário
N100 CTABDEF(Y,X,3,0)	; Início da definição de uma tabela de curvas não periódica com o número 3.
N110 X0 Y0	; 1.Instrução de movimento, define valores iniciais e a 1ª posição de apoio: Valor mestre: 0, valor escravo: 0
N120 X20 Y0	; 2.Posição de apoio: Valor mestre: 0…20, valor escravo: Valor inicial…0

```
Código de programa     Comentário
N130 X100 Y6           ;  3.Posição de apoio:
                          Valor mestre: 20…100, valor escravo: 0…6
N140 X150 Y6           ;  4.Posição de apoio:
                          Valor mestre: 100…150, valor escravo: 6…6
N150 X180 Y0           ;  5.Posição de apoio:
                          Valor mestre: 150…180, valor escravo: 6…0
N200 CTABEND           ;  Fim da definição. A tabela de curvas é gerada em sua
                          representação interna como polinômio de até 5° grau. O
                          cálculo da curva com as posições de apoio indicadas depende
                          do tipo de interpolação modal escolhida (interpolação
                          circular, interpolação linear, interpolação Spline). O
                          estado do programa de peça antes do início da definição é
                          restabelecido.
```

Exemplo 3: Definição de uma tabela de curvas periódica

Definição de uma tabela de curvas periódica com número 2, faixa de valores mestres de 0 até 360, movimento do eixo escravo de 0 para 45 e retorno ao 0:

```
Código de programa                          Comentário
N10 DEF REAL DEPPOS
N20 DEF REAL GRADIENT
N30 CTABDEF(Y,X,2,1)                        ; Início da definição.
N40 G1 X=0 Y=0
N50 POLY
N60 PO[X]=(45.0)
N70 PO[X]=(90.0) PO[Y]=(45.0,135.0,-90)
N80 PO[X]=(270.0)
N90 PO[X]=(315.0) PO[Y]=(0.0,-135.0,90)
N100 PO[X]=(360.0)
N110 CTABEND                                ; Fim da definição.
;Teste da curva através do acoplamento do Y no X:
N120 G1 F1000 X0
N130 LEADON(Y,X,2)
N140 X360
N150 X0
N160 LEADOF(Y,X)

N170 DEPPOS=CTAB(75.0,2,GRADIENT)           ; Leitura da função de tabela com valor
                                              mestre 75.0.
N180 G0 X75 Y=DEPPOS                        ; Posicionamento do eixo mestre e eixo
                                              escravo.
;Após a ativação do acoplamento não é necessária nenhuma sincronização do eixo
escravo.
N190 LEADON(Y,X,2)
N200 G1 X110 F1000
```

```
Código de programa                          Comentário
N210 LEADOF(Y,X)
N220 M30
```

Outras informações

Valor inicial e final da tabela de curvas

Como valor inicial para o início da faixa de definição da tabela de curvas vale a primeira especificação das posições de eixo envolvidas (a primeira instrução de movimento) dentro da definição da tabela de curvas. O valor final da faixa de definição da tabela de curvas é determinado de acordo com o último comando de movimento.

Escopo de linguagem disponível

Na definição da tabela de curvas todo escopo de linguagem NC está disponível.

Indicação

As seguintes especificações não são permitidas em definições de tabelas de curvas:

- Parada de pré-processamento
- Saltos no movimento de eixos mestres (p. ex. na mudança de transformações)
- Uma instrução de movimento exclusivamente para o eixo escravo
- Reversão de movimento do eixo mestre, isto é, a posição do eixo mestre sempre deve ser única
- Instruções `CTABDEF` e `CTABEND` em diferentes níveis de programa.

Efeito de instruções modais

Todas instruções ativas modalmente que existem na definição da tabela de curvas são invalidadas no final da definição de tabelas. Com isso o programa de peça onde se realiza a definição de tabelas encontra-se no mesmo estado antes e após a definição de tabelas.

Atribuições no parâmetro R

As atribuições no parâmetro R dentro da definição de tabelas são desfeitas após o `CTABEND`.

Exemplo:

```
Código de programa                  Comentário
...
R10=5 R11=20                        ; R10=5
...
CTABDEF
G1 X=10 Y=20 F1000
R10=R11+5                           ; R10=25
X=R10
CTABEND
...                                 ; R10=5
```

Ativação do ASPLINE, BSPLINE, CSPLINE

Se em uma definição de tabela de curvas CTABDEF ... CTABEND for ativado um ASPLINE, BSPLINE ou CSPLINE, então deve ser programado pelo menos um ponto de partida antes desta ativação de Spline. Uma ativação imediata após CTABDEF deve ser evitada, caso contrário a Spline da atual posição de eixo será cancelada antes da definição da tabela de curvas.

Exemplo:

Código de programa

```
...
CTABDEF(Y,X,1,0)
X0 Y0
ASPLINE
X=5 Y=10
X10 Y40
...
CTABEND
```

Utilização repetida de tabelas de curvas

A relação funcional do eixo mestre e o eixo escravo, calculada através da tabela de curvas, é mantida no número de tabela selecionado além do fim do programa de peça e além do POWER OFF, isto se a tabela estiver armazenada na memória NC estática (SRAM).

Uma tabela que foi criada na memória dinâmica (DRAM) é deletada com POWER ON e, quando solicitada, deverá ser criada novamente.

A tabela de curvas criada pode ser aplicada em qualquer combinação de eixos mestres e escravos e está em função da configuração de eixos utilizados para criar a tabela de curvas.

Sobregravação de tabelas de curvas

Uma tabela de curvas é sobregravada assim que uma nova definição de tabela utilizar seu número.

Exceção: Uma tabela de curvas está ativa em um acoplamento de eixo ou bloqueada com CTABLOCK.

Indicação

Na sobregravação de tabelas de curvas não é emitido nenhum aviso correspondente!

Definição de tabela de curvas ativa?

Com a variável de sistema $P_CTABDEF pode ser consultado a qualquer momento do programa de peça se uma definição de tabela de curvas está ativa.

Cancelamento da definição de tabela de curvas

O segmento do programa de peça pode ser utilizado novamente como programa de peça real depois de serem retiradas as instruções referentes à definição da tabela de curvas.

Carregamento de tabelas de curvas através da função "Executar externamente"

Para a execução externa de tabelas de curvas deve-se selecionar o buffer de recarga (DRAM) através do MD18360 $MN_MM_EXT_PROG_BUFFER_SIZE de modo que a definição total da tabela de curvas possa ser armazenada ao mesmo tempo no buffer de recarga. Caso contrário a execução do programa de peça é cancelada com um alarme.

Saltos do eixo escravo

Em função do dado de máquina:
MD20900 $MC_CTAB_ENABLE_NO_LEADMOTION
os saltos do eixo escravo podem ser tolerados durante a ausência de movimento do eixo mestre.

9.2.2 Controle de presença de uma tabela de curvas (CTABEXISTS)

Função

Com o comando `CTABEXISTS` pode ser verificado se um determinado número de tabela de curvas está presente na memória NC.

Sintaxe

```
CTABEXISTS(<n>)
```

Significado

`CTABEXISTS`	Verifica se a tabela de curvas de número `<n>` está presente na memória estática ou dinâmica do NC 0 A tabela não existe 1 A tabela existe
`<n>`	Número (**ID**) da tabela de curvas

9.2.3 Eliminação de tabelas de curvas (CTABDEL)

Função

Com o `CTABDEL` pode-se deletar tabelas de curvas.

Indicação

As tabelas de curvas ativadas em um acoplamento de eixos não podem ser deletadas.

Sintaxe

```
CTABDEL(<n>)
CTABDEL(<n>,<m>)
CTABDEL(<n>,<m>,<local de armazenamento>)
CTABDEL( )
CTABDEL(,,<local de armazenamento>)
```

Significado

`CTABDEL`	Comando para deletar tabelas de curvas
`<n>`	Número (**ID**) da tabela de curvas que deve ser deletada Durante a eliminação de uma área de tabela de curvas `CTABDEL(<n>,<m>)` é indicado com `<n>` o número da primeira tabela de curvas da área.
`<m>`	Durante a eliminação de uma área de tabela de curvas `CTABDEL(<n>,<m>)` é indicado com `<m>` o número da última tabela de curvas da área. O `<m>` deve ser maior que o `<n>`!
`<local de armazenamento>`	Especificação do local de armazenamento (opcional) Durante a eliminação **sem** indicação do local de armazenamento as tabelas de curvas indicadas são deletadas da memória estática e dinâmica do NC. Durante a eliminação **com** indicação do local de armazenamento apenas são deletadas as tabelas de curvas presentes na memória indicada. As demais são mantidas. `"SRAM"` Eliminação na memória **estática** do NC `"DRAM"` Eliminação na memória **dinâmica** do NC

Se o `CTABDEL` for programado sem indicação da tabela de curvas que deve ser deletada, então **todas** as tabelas de curvas serão deletadas, inclusive as da memória indicada:

`CTABDEL( )`	Deleta todas as tabelas da memória estática e dinâmica do NC
`CTABDEL(,,"SRAM")`	Deleta todas as tabelas da memória estática do NC
`CTABDEL(,,"DRAM")`	Deleta todas as tabelas da memória dinâmica do NC

Indicação

Se durante a eliminação múltipla `CTABDEL(<n>,<m>)` ou `CTABDEL()` existir pelo menos um acoplamento ativo das tabelas de curvas a serem deletadas, então o comando de eliminação não será executado, isto é, **nenhuma** das tabelas de curvas endereçadas será deletada.

9.2.4 Bloqueio de tabelas de curvas contra eliminação e sobregravação (CTABLOCK, CTABUNLOCK)

Função

As tabelas de curvas podem ser protegidas contra uma eliminação e sobregravação acidental através da definição de um bloqueio. Um bloqueio definido pode ser cancelado a qualquer momento.

Sintaxe

Definição de bloqueio:

```
CTABLOCK(<n>)
CTABLOCK(<n>,<m>)
CTABLOCK(<n>,<m>,<local de armazenamento>)
CTABLOCK()
CTABLOCK(,,<local de armazenamento>)
```

Cancelamento de bloqueio:

```
CTABUNLOCK(<n>)
CTABUNLOCK(<n>,<m>)
CTABUNLOCK(<n>,<m>,<local de armazenamento>)
CTABUNLOCK()
CTABUNLOCK(,,<local de armazenamento>)
```

Significado

CTABLOCK	Comando para **definição** de um bloqueio contra eliminação/sobregravação
CTABUNLOCK	Comando para **cancelamento** de um bloqueio contra eliminação/sobregravação O CTABUNLOCK libera novamente as tabelas bloqueadas com CTABLOCK. As tabelas de curvas que estiverem envolvidas em um acoplamento ativo permanecem bloqueadas e não podem ser deletadas. O bloqueio é cancelado com CTABLOCK assim que o travamento for cancelado com a desativação do acoplamento que estiver ativo. Com isso esta tabela pode ser deletada. Não será mais necessária uma nova chamada do CTABUNLOCK.
<n>	Número (**ID**) da tabela de curvas que deve ser bloqueada/desbloqueada Durante o bloqueio/desbloqueio de uma área de tabela de curvas CTABLOCK(<n>,<m>)/CTABUNLOCK(<n>,<m>) é indicado com <n> o número da primeira tabela de curvas da área.

`<m>`	Durante o bloqueio/desbloqueio de uma área de tabela de curvas `CTABLOCK(<n>,<m>)`/`CTABUNLOCK(<n>,<m>)` é indicado com `<m>` o número da última tabela de curvas da área. O `<m>` deve ser maior que o `<n>`!
`<local de armazenamento>`	Especificação do local de armazenamento (opcional) Durante a definição/cancelamento de um bloqueio **sem** indicação do local de armazenamento as tabelas de curvas indicadas são bloqueadas/desbloqueadas na memória estática e dinâmica do NC. Durante a definição/cancelamento de um bloqueio **com** indicação do local de armazenamento apenas são bloqueadas/desbloqueadas as tabelas de curvas presentes na memória indicada. As demais não são bloqueadas/desbloqueadas. `"SRAM"` Definição/cancelamento de bloqueio na memória **estática** do NC `"DRAM"` Definição/cancelamento de bloqueio na memória **dinâmica** do NC

Se o `CTABLOCK`/`CTABUNLOCK` for programado sem indicação da tabela de curvas que deve ser bloqueada/desbloqueada, então **todas** as tabelas de curvas serão bloqueadas/desbloqueadas, inclusive as da memória indicada:

`CTABLOCK()`	Bloqueia todas as tabelas da memória estática e dinâmica do NC
`CTABLOCK(,,"SRAM")`	Bloqueia todas as tabelas da memória estática do NC
`CTABLOCK(,,"DRAM")`	Bloqueia todas as tabelas da memória dinâmica do NC
`CTABUNLOCK()`	Desbloqueia todas as tabelas da memória estática e dinâmica do NC
`CTABUNLOCK(,,"SRAM")`	Desbloqueia todas as tabelas da memória estática do NC
`CTABUNLOCK(,,"DRAM")`	Desbloqueia todas as tabelas da memória dinâmica do NC

9.2.5 Tabelas de curvas: Determinação de propriedades de tabela (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD)

Função

Com estes comandos podem ser consultadas propriedades importantes de uma tabela de curvas (número de tabela, estado de bloqueio, local de armazenamento, periodicidade).

Sintaxe

```
CTABID(<p>)
CTABID(<p>,<local de armazenamento>)
CTABISLOCK(<n>)
CTABMEMTYP(<n>)
TABPERIOD(<n>)
```

Significado

`CTABID` Fornece o **número de tabela** que está registrado na memória indicada como a `<p>`[a] tabela de curvas.

Exemplo:

O `CTABID(1,"SRAM")` fornece o número da primeira tabela de curvas na memória estática do NC. Neste caso, a primeira tabela de curvas não corresponde ao número de tabela mais alto.

Nota:
Se a ordem das tabelas de curvas é alterada entre duas chamadas consecutivas de `CTABID`, por exemplo, através da eliminação de tabelas de curvas com `CTABDEL`, o `CTABID(<p>,...)` pode fornecer uma tabela diferente da tabela anterior de mesmo número `<p>`.

`CTABISLOCK` Retorna o **estado de bloqueio** da tabela de curvas com o número `<n>`:

0 A tabela não está bloqueada

1 A tabela está bloqueada através do `CTABLOCK`

2 A tabela está bloqueada através de acoplamento ativo

3 A tabela está bloqueada através do `CTABLOCK` e acoplamento ativo

-1 A tabela não existe

`CTABMEMTYP` Fornece o **local de armazenamento** da tabela de curvas com o número `<n>`:

0 Tabela na memória estática do NC

1 Tabela na memória dinâmica do NC

-1 A tabela não existe

`CTABPERIOD` Fornece a **periodicidade** da tabela de curvas com o número `<n>`:

0 A tabela não é periódica

1 A tabela é periódica no eixo mestre

2 A tabela é periódica no eixo mestre e no eixo escravo

-1 A tabela não existe

`<p>` Número de registro na memória

`<n>` Número (**ID**) da tabela de curvas

`<local de armazenamento>` Especificação do local de armazenamento (opcional)

`"SRAM"` Memória **estática** do NC

`"DRAM"` Memória **dinâmica** do NC

Nota:
Se não for programado nenhum valor para este parâmetro, então é utilizado o local de armazenamento padrão definido no dado MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE.

9.2.6 Leitura dos valores da tabela de curvas (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX)

Função

Os seguintes valores de tabela de curvas podem ser lidos no programa de peça:

- Valores de eixo escravo e eixo mestre no início e no fim de uma tabela de curvas
- Valores de eixo escravo no início e no fim de um segmento de curva
- Valor de eixo escravo para um valor de eixo mestre
- Valor de eixo mestre para um valor de eixo escravo
- Valor mínimo e valor máximo do eixo escravo
  - na faixa de definição inteira da tabela de curvas

    ou
  - em um intervalo definido da tabela de curvas

Sintaxe

```
CTABTSV(<n>,<gradiente>[,<eixo escravo>])
CTABTEV(<n>,<gradiente>[,<eixo escravo>])
CTABTSP(<n>,<gradiente>[,<eixo mestre>])
CTABTEP(<n>,<gradiente>[,<eixo mestre>])
CTABSSV(<valor mestre>,<n>,<gradiente>[,<eixo escravo>])
CTABSEV(<valor mestre>,<n>,<gradiente>[,<eixo escravo>])
CTAB(<valor mestre>,<n>,<gradiente>[,<eixo escravo>,<eixo mestre>]
CTABINV(<valor escravo>,<valor aproximado>,<n>,<gradiente>[,<eixo
escravo>,<eixo mestre>]
CTABTMIN(<n>[,<eixo escravo>])
CTABTMAX(<n>[,<eixo escravo>])
CTABTMIN(<n>,<a>,<b>[,<eixo escravo>,<eixo mestre>])
CTABTMAX(<n>,<a>,<b>[,<eixo escravo>,<eixo mestre>])
```

Significado

CTABTSV:	Leitura do valor de eixo escravo no **início** da tabela de curvas nº <n>
CTABTEV:	Leitura do valor de eixo escravo no **fim** da tabela de curvas nº <n>
CTABTSP:	Leitura do valor de eixo mestre no **início** da tabela de curvas nº <n>
CTABTEP:	Leitura do valor de eixo mestre no **fim** da tabela de curvas nº <n>
CTABSSV:	Leitura do valor de eixo escravo no **início** do segmento de curva pertencente ao valor de eixo mestre especificado (<valor mestre>)
CTABSEV:	Leitura do valor de eixo escravo no **fim** do segmento de curva pertencente ao valor de eixo mestre especificado (<valor mestre>)
CTAB:	Leitura do valor de eixo escravo para o valor de eixo mestre especificado (<valor mestre>)
CTABINV:	Leitura do valor de eixo mestre para o valor de eixo escravo especificado (<valor escravo>)
CTABTMIN:	Determinação do **valor mínimo** do eixo escravo: • na faixa de definição inteira da tabela de curvas ou • em um intervalo definido <a> ... <b>
CTABTMAX:	Determinação do **valor máximo** do eixo escravo: • na faixa de definição inteira da tabela de curvas ou • em um intervalo definido <a> ... <b>
<n>:	Número (**ID**) da tabela de curvas
<gradiente>:	No parâmetro <gradiente> é retornada a **inclinação** da função de tabela de curvas na posição determinada
<eixo escravo>:	Eixo, cujo movimento deve ser calculado através da tabela de curvas (opcional)
<eixo mestre>:	Eixo, que fornece os valores mestres para o cálculo do movimento do eixo escravo (opcional)
<valor escravo>:	Valor de eixo escravo para leitura do valor de eixo mestre correspondente com CTABINV
<valor mestre>:	Valor de eixo mestre: • para leitura do valor de eixo escravo correspondente com CTAB ou • para a seleção do segmento de curvas com CTABSSV/CTABSEV
<valor aproximado>:	A associação de um valor de eixo mestre com um valor de eixo escravo com CTABINV nem sempre precisa ser única. Por isso o CTABINV requer um valor aproximado como parâmetro para o valor de eixo mestre esperado.
<a>:	Limite **inferior** do intervalo de valor mestre com CTABTMIN/CTABTMAX
<b>:	Limite **superior** do intervalo de valor mestre com CTABTMIN/CTABTMAX **Nota:** O intervalo de valores mestres <a> ... <b> deve estar dentro da faixa de definição da tabela de curvas.

Exemplos

Exemplo 1:

Determinação dos valores de eixo escravo e valores de eixo mestre no início e no fim da tabela de curvas, assim como do valor mínimo e do valor máximo do eixo escravo na faixa de definição inteira da tabela de curvas.

```
Código de programa                          Comentário
N10 DEF REAL STARTPOS
N20 DEF REAL ENDPOS
N30 DEF REAL STARTPARA
N40 DEF REAL ENDPARA
N50 DEF REAL MINVAL
N60 DEF REAL MAXVAL
N70 DEF REAL GRADIENT
...
N100 CTABDEF(Y,X,1,0)                       ; Início da definição de tabela
N110 X0 Y10                                 ; Posição de partida do 1° segmento de tabela
N120 X30 Y40                                ; Posição final do 1° segmento de tabela = Posição de
                                              partida do 2° segmento de tabela
N130 X60 Y5                                 ; Posição final do 2° segmento de tabela = ...
N140 X70 Y30
N150 X80 Y20
N160 CTABEND                                ; Fim da definição de tabela.
...
N200 STARTPOS=CTABTSV(1,GRADIENT)           ; Valor de eixo escravo no início da tabela de curvas =
                                              10
N210 ENDPOS=CTABTEV(1,GRADIENT)             ; Valor de eixo escravo no fim da tabela de curvas = 20
N220 STARTPARA=CTABTSP(1,GRADIENT)          ; Valor de eixo mestre no início da tabela de curvas = 0
N230 ENDPARA=CTABTEP(1,GRADIENT)            ; Valor de eixo mestre no fim da tabela de curvas = 80
N240 MINVAL=CTABTMIN(1)                     ; Valor mínimo do eixo escravo com Y=5
N250 MAXVAL=CTABTMAX(1)                     ; Valor máximo do eixo escravo com Y=40
```

Exemplo 2:

Determinação dos valores de eixo escravo no início e no fim do segmento de curva pertencente ao valor de eixo mestre X=30.

```
Código de programa                          Comentário
N10 DEF REAL STARTPOS
N20 DEF REAL ENDPOS
N30 DEF REAL GRADIENT
...
N100 CTABDEF(Y,X,1,0)                       ; Início da definição de tabela.
N110 X0 Y0                                  ; Posição de partida do 1° segmento de tabela
N120 X20 Y10                                ; Posição final do 1° segmento de tabela = Posição de
```

```
Código de programa                                  Comentário
                                                      partida do 2° segmento de tabela
N130 X40 Y40                                          Posição final do 2° segmento de tabela = ...
N140 X60 Y10
N150 X80 Y0
N160 CTABEND                                        ; Fim da definição de tabela.
...
N200 STARTPOS=CTABSSV(30.0,1,GRADIENT)              ; Posição de partida Y no 2° segmento = 10
N210 ENDPOS=CTABSEV(30.0,1,GRADIENT)                ; Posição final Y no 2° segmento = 40
```

Outras informações

Utilização em ações síncronas

Todos comandos para leitura de valores de tabelas de curvas também podem ser utilizados em ações síncronas (veja também o capítulo "Ações síncronas de movimentos").

Para utilização dos comandos `CTABINV`, `CTABTMIN` e `CTABTMAX` deve-se prestar atenção para que:

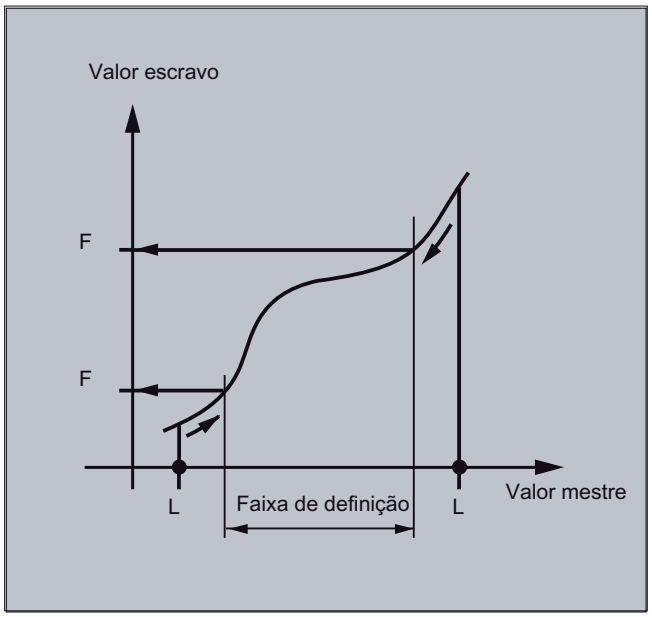
- no momento de execução uma capacidade suficiente do NC esteja disponível

  ou

- o número de segmentos da tabela de curvas seja consultado antes da chamada para, eventualmente, dividir a tabela afetada

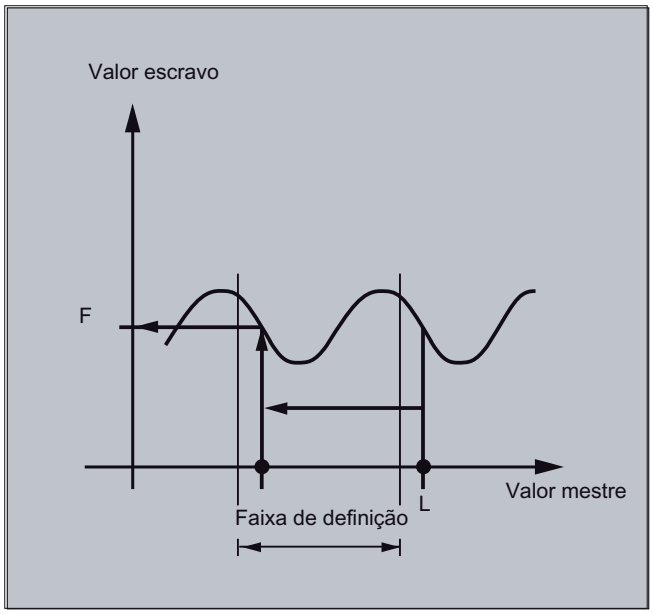
CTAB em tabelas de curvas não periódicas

Se o `<valor mestre>` especificado estiver fora da faixa de definição, é retornado o limite superior ou inferior como valor escravo.



CTAB em tabelas de curvas periódicas

Se o <valor mestre> especificado estiver fora da faixa de definição, é avaliado o valor mestre do Modulo da faixa de definição e retornado o valor escravo correspondente:



Valor aproximado para CTABINV

O comando CTABINV requer um valor aproximado para o valor mestre esperado. O CTABINV retorna o valor mestre que está mais próximo do valor aproximado. Por exemplo, o valor aproximado pode ser o valor mestre de um ciclo de interpolação anterior.

Inclinação da função de tabela de curvas

A emissão da inclinação (<gradiente>) permite o cálculo da velocidade do eixo mestre ou eixo escravo na respectiva posição.

Especificação do eixo mestre ou eixo escravo

A especificação opcional do eixo mestre ou eixo escravo é importante se o eixo mestre e o eixo escravo estiverem configurados em diferentes unidades lineares.

CTABSSV, CTABSEV

Os comandos CTABSSV e CTABSEV nos seguintes casos **não** são adequadospara a consulta dos segmentos programados:

- Foram programados círculos e evolventes.
- Chanfros e arredondamentos estão ativos com CHF/RND.
- A suavização está ativa com G643.
- O compressor de blocos NC está ativo com COMPON/COMPCURV/COMPCAD.

9.2.7 Tabelas de curvas: Controle do aproveitamento de recursos (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL)

Função

Com estes comandos o programador tem a possibilidade de se manter atualizado com informações sobre a ocupação dos recursos para tabelas de curvas, segmentos de tabelas e polinômios.

Sintaxe

```
CTABNO
CTABNOMEM(<local de armazenamento>)
CTABFNO(<local de armazenamento>)
CTABSEGID(<n>,<local de armazenamento>)
CTABSEG(<local de armazenamento>,<tipo de segmento>)
CTABFSEG(<local de armazenamento>,<tipo de segmento>)
CTABMSEG(<local de armazenamento>,<tipo de segmento>)
CTABPOLID(<n>)
CTABPOL(<local de armazenamento>)
CTABFPOL(<local de armazenamento>)
CTABMPOL(<local de armazenamento>)
```

Significado

`CTABNO`	Determinação do número total das tabelas de curvas **definidas** (na memória estática e dinâmica do NC)
`CTABNOMEM`	Determinação das tabelas de curvas **definidas** no `<local de armazenamento>` indicado
`CTABFNO`	Determinação das tabelas de curvas **ainda possíveis** no `<local de armazenamento>` indicado
`CTABSEGID`	Determinação dos segmentos de curva do `<tipo de segmento>` indicado, que são utilizados pela tabela de curvas de número `<n>`
`CTABSEG`	Determinação dos segmentos de curva **utilizados** do `<tipo de segmento>` indicado no `<local de armazenamento>` indicado
`CTABFSEG`	Determinação dos segmentos de curva **ainda possíveis** do `<tipo de segmento>` indicado no `<local de armazenamento>` indicado
`CTABMSEG`	Determinação dos segmentos de curva **máximos possíveis** do `<tipo de segmento>` indicado no `<local de armazenamento>` indicado
`CTABPOLID`	Determinação da quantidade de polinômios de curvas, que são utilizados pela tabela de curvas de número `<n>`
`CTABPOL`	Determinação da quantidade de polinômios de curvas **utilizados** no `<local de armazenamento>` indicado
`CTABFPOL`	Determinação da quantidade de polinômios de curvas **ainda possível** no `<local de armazenamento>` indicado

`CTABMPOL` Determinação da quantidade de polinômios de curvas **máxima possível** no `<local de armazenamento>` indicado

`<n>` Número (**ID**) da tabela de curvas

`<local de armazenamento>` Especificação do local de armazenamento (opcional)

- `"SRAM"` Memória **estática** do NC
- `"DRAM"` Memória **dinâmica** do NC

Nota:
Se não for programado nenhum valor para este parâmetro, então é utilizado o local de armazenamento padrão definido no dado MD20905 $MC_CTAB_DEFAULT_MEMORY_TYPE.

`<tipo de segmento>` Especificação do tipo de segmento (opcional)

- `"L"` Segmentos lineares
- `"P"` Segmentos de polinômio

Nota:
Se não for programado nenhum valor para este parâmetro, então é retornada a soma dos segmentos lineares e de polinômio.

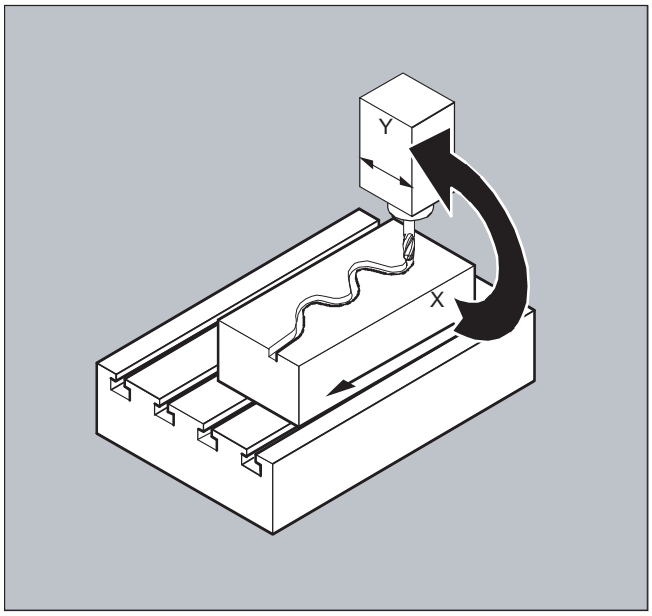
9.3 Acoplamento axial de valor mestre (LEADON, LEADOF)

Indicação

Esta função não está disponível para o SINUMERIK 828D!

Função

No acoplamento axial de valor mestre o eixo mestre e o eixo escravo são movimentados de forma sincronizada. Neste caso a respectiva posição do eixo escravo é atribuída para uma posição, eventualmente simulada, do eixo mestre através de uma tabela de curvas e de um polinômio calculado a partir desta.



O **eixo mestre** é o eixo que fornece os valores de entrada para a tabela de curvas. O **eixo escravo** é o eixo que assume as posições calculadas na tabela de curvas.

Acoplamento de valor real e valor nominal

Como valores mestres, portanto valores de saída para determinar a posição do eixo escravo, podem ser utilizados:

- Valores reais da posição do eixo mestre: Acoplamento de valor real
- Valores reais da posição do eixo mestre: Acoplamento do valor nominal

O acoplamento de valor mestre sempre é aplicado no sistema de coordenadas básico.

Sobre a criação de tabelas de curvas, veja o capítulo "Tabelas de curvas".
Sobre acoplamento de valor mestre, veja /FB/, M3, Movimento acoplado e acoplamento de valor mestre.

Sintaxe

```
LEADON(EixoE,EixoM,n)
LEADOF(EixoE,EixoM)
```

ou desativação sem indicação do eixo mestre:

```
LEADOF(EixoE)
```

O acoplamento de valor mestre pode ser ativado e desativado tanto a partir do programa de peças como durante o movimento de ações sincronizadas, veja o capítulo "Ações sincronizadas de movimentos".

Significado

`LEADON`	Ativação do acoplamento de valor mestre
`LEADOF`	Desativação do acoplamento de valor mestre
`EixoE`	Eixo escravo
`EixoM`	Eixo mestre
`n`	Número da tabela de curvas
`$SA_LEAD_TYPE`	Comutação entre acoplamento de valor nominal e valor real

Desativar o acoplamento de valor mestre, LEADOF

Com a desativação do acoplamento de valor mestre o eixo escravo torna-se novamente um eixo normal de comando!

Acoplamento axial de valor mestre e diversos estados operacionais, RESET

Independentemente do ajuste no dado de máquina, os acoplamentos de valor mestre são desativados com RESET.

Exemplo de acoplamento de valor mestre a partir da ação sincronizada

Em uma prensa um acoplamento mecânico convencional entre um eixo mestre (eixo de estampagem) e os eixos de um sistema Transfer dotado de eixos Transfer e eixos auxiliares deve ser substituído por um sistema de acoplamento eletrônico.

Aqui demonstra-se como se substitui um sistema Transfer mecânico por um sistema Transfer eletrônico em uma prensa. Os processos de acoplamento e desacoplamento são realizados como **ações sincronizadas estáticas**.

A partir do eixo mestre de valor mestre (eixo de estampagem) os eixos Transfer e eixos auxiliares são definidos como eixos escravos através de tabelas de curvas.

Eixos escravos

X - Eixo de avanço e longitudinal
YL - Eixo de fechamento e eixo transversal
ZL - Eixo de elevação
U - Avanço de rolo, eixo auxiliar
V - cabeçote de guia, eixo auxiliar
W - Aplicação de graxa, eixo auxiliar

Ações

As ações ocorrem, por exemplo, em ações sincronizadas:

- Acoplamento, `LEADON(eixo escravo, eixo mestre, número da tabela de curvas)`
- Desacoplamento, `LEADOF(eixo escravo, eixo mestre)`
- Definição do valor real, `PRESETON(eixo, valor)`
- Definir marcador, `$AC_MARKER[i]= valor`
- Tipo de acoplamento: valor mestre real/virtual
- Aproximação de posições de eixo, `POS[eixo]=valor`

Condições

Como condições são avaliadas as entradas digitais rápidas, variáveis de tempo real `$AC_MARKER` e comparações de posição que estiverem interligadas com o operador lógico AND.

Indicação

No capítulo a seguir foram utilizadas quebras de linhas, recuos e parágrafos em **negrito** especialmente para facilitar a leitura da programação. Para o comando tudo que estiver em um número de linha constitui uma linha separada.

Comentário

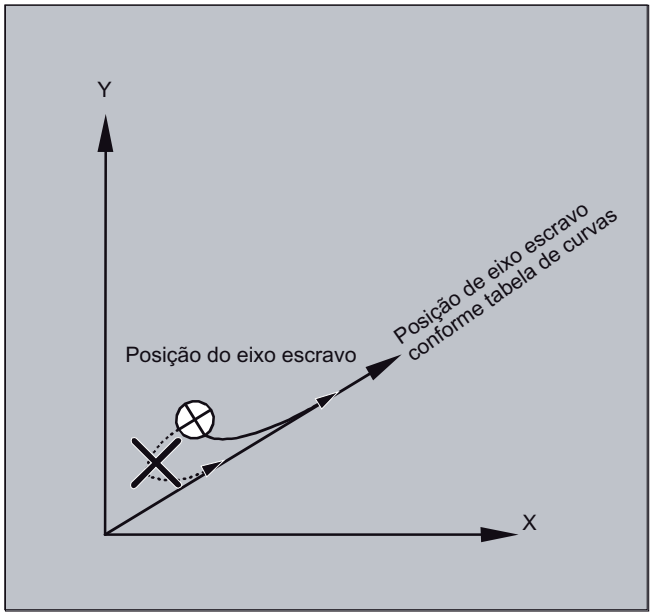
Código de programa	**Comentário**
	; Define todas ações sincronizadas estáticas.
	; ****Resetar marcador
N2 $AC_MARKER[0]=0 $AC_MARKER[1]=0 $AC_MARKER[2]=0 $AC_MARKER[3]=0 $AC_MARKER[4]=0 $AC_MARKER[5]=0 $AC_MARKER[6]=0 $AC_MARKER[7]=0	
	; **** E1 0=>1 acoplamento Transfer ON
N10 IDS=1 EVERY ($A_IN[1]==1) AND ($A_IN[16]==1) AND ($AC_MARKER[0]==0) DO LEADON(X,valor mestre,1) LEADON(YL,valor mestre,2) LEADON(ZL,valor mestre,3) $AC_MARKER[0]=1	
	; **** E1 0=>1 acoplamento do avanço de rolos ON
N20 IDS=11 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_MARKER[5]==0) DO LEADON(U,valor mestre,4) PRESETON(U,0) $AC_MARKER[5]=1	
	; **** E1 0->1 acoplamento do cabeçote guia ON
N21 IDS=12 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_MARKER[6]==0) DO LEADON(V,valor mestre,4) PRESETON(V,0) $AC_MARKER[6]=1	
	; **** E1 0->1 acoplamento da aplicação de graxa ON
N22 IDS=13 EVERY ($A_IN[1]==1) AND ($A_IN[5]==0) AND ($AC_MARKER[7]==0) DO LEADON(W,valor mestre,4) PRESETON(W,0) $AC_MARKER[7]=1	
	; **** E2 0=>1 acoplamento OFF
N30 IDS=3 EVERY ($A_IN[2]==1) DO LEADOF(X,valor mestre) LEADOF(YL,valor mestre) LEADOF(ZL,valor mestre) LEADOF(U,valor mestre) LEADOF(V,valor mestre) LEADOF(W,valor mestre) $AC_MARKER[0]=0 $AC_MARKER[1]=0 $AC_MARKER[3]=0 $AC_MARKER[4]=0 $AC_MARKER[5]=0 $AC_MARKER[6]=0 $AC_MARKER[7]=0	
....	
N110 G04 F01	
N120 M30	

Descrição

O acoplamento de valor mestre requer a sincronização do eixo mestre e do eixo escravo. Esta sincronização somente pode ser alcançada se o eixo escravo estiver dentro da faixa de tolerância da curva calculada pela tabela de curvas durante a ativação do acoplamento do valor mestre.

A faixa de tolerância para a posição do eixo escravo é definida pelo dado de máquina `MD 37200: COUPLE_POS_POL_COARSE A_LEAD_TYPE .`

Se com a ativação do acoplamento de valor mestre o eixo escravo ainda não estiver na respectiva posição, então o processo sincronizado será realizado automaticamente assim que o valor nominal de posição calculado para o eixo escravo realmente se aproximar da posição do eixo escravo. Neste caso o eixo escravo é movimentado durante o processo de sincronização para o sentido que foi definido pela velocidade nominal do eixo escravo (calculado a partir da velocidade do eixo mestre e conforme a tabela de curvas CTAB).

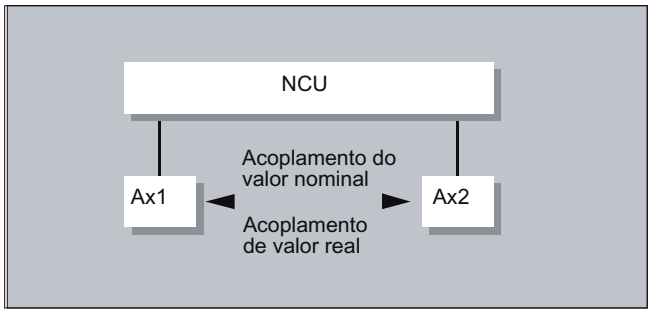


Sem sincronismo

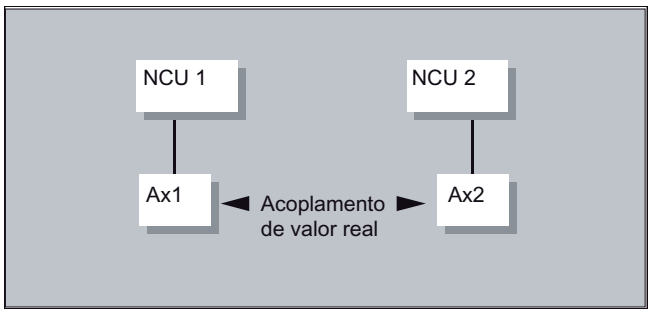
Se com a ativação do acoplamento de valor mestre a posição nominal calculada se afastar da atual posição de eixo escravo, não será realizada nenhuma sincronização.

Acoplamento de valor real e valor nominal

Quando comparado ao acoplamento de valor real, o acoplamento de valor nominal oferece uma melhor sincronização entre o eixo mestre e o eixo escravo e por isso está ajustado como padrão.



O acoplamento de valor nominal somente é possível se o eixo mestre e o eixo escravo forem interpolados pela mesma NCU. No caso de um eixo mestre externo, o eixo escravo apenas pode ser acoplado no eixo mestre através de valores reais.



Uma **comutação** somente é possível através do dado de ajuste `$SA_LEAD_TYPE`.

A comutação entre o acoplamento de valor real e o de valor nominal sempre deve ser realizado com o eixo escravo imobilizado. Pois apenas em estado parado é realizada uma nova sincronização após a comutação.

Exemplo de aplicação

A leitura dos valores reais não pode ser executada isenta de falhas com grandes vibrações da máquina. Na aplicação do acoplamento do valor mestre em uma linha Transfer de prensas, para os passos de trabalho de maior vibração pode ser necessário comutar do acoplamento de valor real para o acoplamento de valor nominal.

Simulação do valor mestre no acoplamento de valor nominal

O interpolador do eixo mestre pode ser separado do servo através de dado de máquina. Desse modo no acoplamento do valor nominal podem ser gerados valores nominais sem movimento efetivo do eixo mestre.

Os valores mestres gerados através do acoplamento de valor nominal podem ser lidos a partir das seguintes variáveis para utilização em ações sincronizadas:

- $AA_LEAD_P	Posição de valor mestre
- $AA_LEAD_V	Velocidade de valor mestre

Gerar valores mestres

Os valores mestres também podem ser gerados com outros procedimentos próprios programados. Os valores mestres gerados são gravados na variável

- $AA_LEAD_SP	Posição de valor mestre
- $AA_LEAD_SV	Velocidade de valor mestre

e também lidos da própria. Para a utilização destas variáveis deve-se definir o dado de ajuste`$SA_LEAD_TYPE = 2`.

Estado do acoplamento

No programa de peça NC se pode consultar o estado do acoplamento com as seguintes variáveis de sistema:

`$AA_COUP_ACT[eixo]`
0: Nenhum acoplamento ativo
16: Acoplamento de valor mestre ativo

Gerenciamento de estado em ações sincronizadas

Processos de ativação e de acoplamento são gerenciados através das variáveis de tempo real:

`$AC_MARKER[i] = n`
com:
número de marcador i
valor de estado n

9.4 Caixa de transmissão eletrônica (EG)

Função

Com o auxílio da função "caixa de transmissão eletrônica" é possível controlar o movimento de um **eixo escravo** conforme o bloco de movimento linear em função de até cinco **eixos mestres**. As relações entre os eixos mestres e o eixo escravo são definidas para cada eixo mestre através do fator de acoplamento.

A proporção de movimento do eixo escravo é formada mediante a adição com base nas proporções de movimento dos eixos mestres multiplicada pelos fatores de acoplamento correspondentes. Com a ativação de um grupo de eixos EG a sincronização do eixo escravo pode ser forçada para uma posição definida. A partir do programa de peça um grupo de caixa de transmissão pode ser:

- definido,
- ativado,
- desativado,
- deletado

.

O movimento do eixo escravo pode ser, alternativamente, derivado dos

- valores nominais do eixos mestres assim como os
- valores reais dos eixos mestres.

Como ampliação também se pode realizar relações entre os eixos mestres e o eixo escravo mediante **tabelas de curvas** (veja o capítulo sobre comportamento de percurso). As caixas de transmissão eletrônicas podem ser conectadas em cascata, isto é, o eixo escravo de uma caixa de transmissão eletrônica pode ser o eixo mestre de outra caixa de transmissão eletrônica.

9.4.1 Definir caixa de transmissão eletrônica (EGDEF)

Função

Um grupo de eixos EG é definido com a especificação do eixo escravo, por pelo menos um, mas no máximo, cinco eixos mestres com o tipo de acoplamento correspondente.

Requisitos

Condição para uma definição de grupo de eixos EG:

Para o eixo escravo ainda não pode ser definido nenhum acoplamento de eixo (eventualmente, um existente deve ser deletado antes com `EGDEL`).

Sintaxe

```
EGDEF(eixo escravo,eixo mestre1,tipo de acomplamento1,eixo
mestre2,tipo de acoplamento2,...)
```

Significado

`EGDEF`	Definição de uma caixa de transmissão eletrônica
`Eixo escravo`	Eixo que é controlado por eixos mestres
`Eixo mestre1 ,...,Eixo mestre5`	Eixos que controlam os eixos escravos
`Tipo de acoplamento1 ,...,Tipo de acoplamento5`	Tipo de acoplamento O tipo de acoplamento deve ser o mesmo para todos eixos mestres e por isso que deve ser especificado individualmente para cada eixo mestre.

Valor:	Significado:
0	O eixo escravo é controlado pelo **valor real** do eixo mestre correspondente.
1	O eixo escravo é controlado pelo **valor nominal** do eixo mestre correspondente.

Indicação

Os fatores de acoplamento são pré-definidos com zero na definição do grupo de acoplamento EG.

Indicação

O `EGDEF` ativa a parada de pré-processamento. A definição da caixa de transmissão com `EGDEF` também pode ser utilizada sem alteração se atua um ou vários eixos mestres através da **tabela de curvas** no eixo escravo.

Exemplo

Código de programa	Comentário
`EGDEF(C,B,1,Z,1,Y,1)`	`; Definição de um grupo de eixos EG. Os eixos mestres B, Z e Y controlam o eixo escravo C através do valor nominal.`

9.4.2 Ativar caixa de transmissão eletrônica (EGON, EGONSYN, EGONSYNE)

Função

Para a ativação de um grupo de eixos EG existem 3 variantes.

Sintaxe

Variante 1:

O grupo de eixos EG é ativado seletivamente sem sincronização com:

```
EGON(FA,"modo de mudança de
blocos",LA1,Z1,N1,LA2,Z2,N2,...,LA5,Z5,N5)
```

Variante 2:

O grupo de eixos EG é ativado seletivamente com sincronização com

```
EGONSYN(FA,"modo de mudança de
eixos",SynPosFA,[,LAi,SynPosLAi,Zi,Ni])
```

Variante 3:

O grupo de eixos EG é ativado seletivamente com sincronização e especificado o modo de aproximação com:

```
EGONSYNE(FA,"modo de mudança de blocos",SynPosFA,modo de
aproximação[,LAi,SynPosLAi,Zi,Ni])
```

Significado

Variante 1:

`FA`	Eixo escravo	
`Modo de mudança de blocos`	Podem ser utilizados os seguintes modos:	
	`"NOC"`	A mudança de blocos é realizada imediatamente
	`"FINE"`	A mudança de blocos é realizada com "sincronização fina"
	`"COARSE"`	A mudança de blocos é realizada com "sincronização aproximada"
	`"IPOSTOP"`	A mudança de blocos é realizada com sincronização de valor nominal
`LA1, ... LA5`	Eixos mestres	
`Z1, ... Z5`	Numerador para o fator de acoplamento i	
`N1, ... N5`	Denominador para o fator de acoplamento i	
	Fator de acoplamento i = numerador i / denominador i	

Somente podem ser programados os eixos mestres que foram previamente especificados com `EGDEF`. Deve ser programado pelo menos um eixo mestre.

Variante 2:

`FA`	Eixo escravo
`Modo de mudança de blocos`	Podem ser utilizados os seguintes modos:
	`"NOC"` A mudança de blocos é realizada imediatamente
	`"FINE"` A mudança de blocos é realizada com "sincronização fina"
	`"COARSE"` A mudança de blocos é realizada com "sincronização aproximada"
	`"IPOSTOP"` A mudança de blocos é realizada com sincronização de valor nominal
`[,LAi,SynPosLAi,Zi,Ni]`	(não escrever colchetes)
	Mín. 1, máx. 5 seqüências de:
`LA1, ... LA5`	Eixos mestres
`SynPosLAi`	Posição sincronizada para o eixo mestre i
`Z1, ... Z5`	Numerador para o fator de acoplamento i
`N1, ... N5`	Denominador para o fator de acoplamento i
	Fator de acoplamento i = numerador i / denominador i

Somente podem ser programados eixos mestres que foram previamente especificados com `EGDEF`. Com as "posições sincronizadas" programadas para o eixo escravo (`SynPosFA`) e para os eixos mestres (`SynPosLA`) são definidas posições onde o grupo de acoplamento é considerado como *sincronizado*. Se a caixa de transmissão eletrônica não estiver em estado síncrono ao ser ativada, então o eixo escravo movimenta-se até sua posição sincronizada definida.

Variante 3:

Os parâmetros correspondem aos da variante 2 além de:

`Modo de aproximação`	Podem ser utilizados os seguintes modos:
	`"NTGT"` Aproximar o próximo vão entre dentes com otimização de tempo
	`"NTGP"` Aproximar o próximo vão entre dentes com otimização de curso
	`"ACN"` Deslocar o eixo rotativo no sentido de giro negativo e de modo absoluto
	`"ACP"` Deslocar o eixo rotativo no sentido de giro positivo e de modo absoluto
	`"DCT"` Com otimização de tempo para a posição sincronizada programada
	`"DCP"` Com otimização de curso para a posição sincronizada programada

A variante 3 apenas atua em eixos escravos Modulo que estão acoplados em eixos mestres Modulo. A otimização de tempo considera os limites de velocidade do eixo escravo.

Outras informações

Descrição das variantes de ativação

Variante 1:

No momento da ativação, as posições dos eixos mestres assim como o eixo escravo são armazenados como "posições sincronizadas". As "posições sincronizadas" podem ser lidas com a variável de sistema `$AA_EG_SYN`.

Variante 2:

Quando os eixos Modulo estiverem no grupo de acoplamento, seus valores de posição serão reduzidos no modulo. Desse modo se garante uma aproximação mais rápida possível da posição sincronizada (chamada *sincronização relativa*: p. ex. o próximo vão entre dentes). Se para o eixo escravo não for especificado o sinal de interface "liberação da sobreposição do eixo escravo" DB(30 +número de eixo), DBX 26 Bit 4, então não será realizado o deslocamento até a posição sincronizada. Em vez disso, o programa é parado no bloco EGONSYN e emitida a mensagem de alarme 16771 auto-deletável até que o sinal acima mencionado seja ativado.

Variante 3:

A distância dos dentes (graus) resulta de: 360 * Zi/Ni. Para o caso em que o eixo escravo estiver parado no momento da chamada, a otimização de curso oferece o mesmo comportamento como na otimização de tempo.

Se o eixo escravo estiver em marcha, se realiza a sincronização com `NTGP`, independentemente da velocidade atual do eixo escravo, até o próximo vão entre dentes. Se o eixo escravo estiver em marcha, se realiza a sincronização com `NTGT`, dependentemente da velocidade atual do eixo escravo, até o próximo vão entre dentes. Com esse propósito o eixo também será freado quando necessário.

Tabelas de curvas

Se para um dos eixos mestres for utilizada uma **tabela de curvas**, então deve-se:

Ni	passar para 0 o denominador do fator de acoplamento para acoplamentos lineares. (o denominador 0 não seria permitido para acoplamentos lineares). O denominador zero indica para o comando que
Zi	se deve interpretar como número da tabela de curvas a ser utilizada. A tabela de curvas com o número indicado deve estar definida no momento da ativação.
LAi	A especificação do eixo mestre corresponde à especificação do eixo mestre no caso de acoplamento com fator de acoplamento (acoplamento linear).

Para mais informações sobre a utilização de tabelas de curvas e a conexão em cascata de caixas de transmissão eletrônicas e suas sincronização, veja na:
Literatura:
Manual de funções especiais; Acoplamentos de eixo e ESR (M3), Capítulo "Movimento acoplado e acoplamento de valor mestre".

Comportamento da caixa de transmissão eletrônica com Power On, RESET, mudança de modo de operação, localização

- Após Power On **nenhum** acoplamento está ativo.
- Os acoplamentos ativos são mantidos mesmo com RESET e a mudança de modo de operação.
- Na localização de blocos os comandos de ativação, deletação, definição da caixa de transmissão eletrônica não são executados nem coletados, mas ignorados.

Variáveis de sistema da caixa de transmissão eletrônica

Com o auxílio de variáveis de sistema da transmissão eletrônica, o programa de peça pode verificar, e se for necessário, reagir aos estados do grupo de eixos EG.

As variáveis de sistema da caixa de transmissão eletrônica são identificadas da seguinte forma:

$AA_EG_ ...

ou

$VA_EG_ ...

Literatura:
Manual das variáveis de sistema

9.4.3 Desativar transmissão eletrônica (EGOFS, EGOFC)

Função

Para a desativação de um grupo de eixos EG existem 3 variantes.

Programação

Variante 1:

Sintaxe	Significado
`EGOFS(eixo escravo)`	A caixa de transmissão eletrônica é desativada. O eixo escravo é desacelerado até parar. A chamada ativa a parada de pré-processamento.

Variante 2:

Sintaxe	Significado
`EGOFS(eixo escravo,eixo mestre1,…,eixo mestre5)`	Esta parametrização do comando permite eliminar **seletivamente** a influência que exerce cada eixo mestre sobre o movimento do eixo escravo.

Deve-se especificar pelo menos um eixo mestre. A influência dos eixos mestres especificados sobre o eixo escravo é desativada de forma controlada. A chamada ativa a parada de pré-processamento. Se ainda permanecerem eixos mestres ativos, então o eixo escravo continua em marcha sob esta influência. Uma vez eliminadas todas influências de eixos mestres desta forma, então o eixo escravo será desacelerado até parar.

Variante 3:

Sintaxe	Significado
`EGOFC(fuso escravo1)`	A caixa de transmissão eletrônica é desativada. O eixo escravo continua em marcha com a rotação e velocidade ativas no momento da desativação. A chamada ativa a parada de pré-processamento.

Indicação

Esta função somente é permitida para fusos.

9.4.4 Deletar a definição de uma caixa de transmissão eletrônica (EGDEL)

Função

Um grupo de eixos EG deve estar desativado antes de sua definição ser deletada.

Programação

Sintaxe	Significado
`EGDEL(eixo escravo)`	A definição de acoplamento do grupo de eixos é deletada. Até alcançar o número máximo de grupos de eixos ativos simultaneamente pode-se definir novamente outros grupo de eixos com `EGDEF`. A chamada ativa a parada de pré-processamento.

9.4.5 Avanço por rotação (G95) / caixa de transmissão eletrônica (FPR)

Função

Com o comando `FPR` também se pode especificar o eixo escravo de uma transmissão eletrônica como eixo determinante para o avanço por rotação. Para este caso se aplica o seguinte comportamento:

- O avanço depende da velocidade nominal do eixo escravo da caixa de transmissão eletrônica.
- A velocidade nominal é calculada a partir das velocidades dos fusos mestres e eixos mestres Modulo (que não são eixos de percurso) e seus fatores de acoplamento associados.
- Não são considerados os componentes de velocidade de eixos lineares e eixos que não são Modulo e os movimentos sobrepostos do eixo escravo.

9.5 Fuso sincronizado

Função

No modo sincronizado existe um fuso mestre (LS) e um fuso escravo (FS), o assim chamado **par de fusos sincronizados**. Com o acoplamento ativo (modo sincronizado) o fuso escravo acompanha os movimentos do fuso mestre de acordo com a relação de funcionamento definida.

O par de fusos sincronizados pode ser configurado de forma permanente para qualquer máquina através dos dados de máquina específicos de canal, ou definido pelo usuário através do programa de peça CNC. Por canal NC podem ser definidos até 2 pares de fusos sincronizados simultaneamente.

A partir do programa de peça um grupo de caixa de transmissão pode ser

- definido e alterado
- ativado
- desativado
- deletado

.

Além disso, em função do software se pode

- esperar pela condição de sincronismo
- alterar o comportamento da mudança de blocos
- selecionar o tipo de acoplamento entre nominal ou real ou especificar o deslocamento angular entre fuso mestre e fuso escravo
- assumir uma programação especificada do fuso escravo ao ativar o acoplamento
- corrigir um desvio de sincronismo medido ou conhecido.

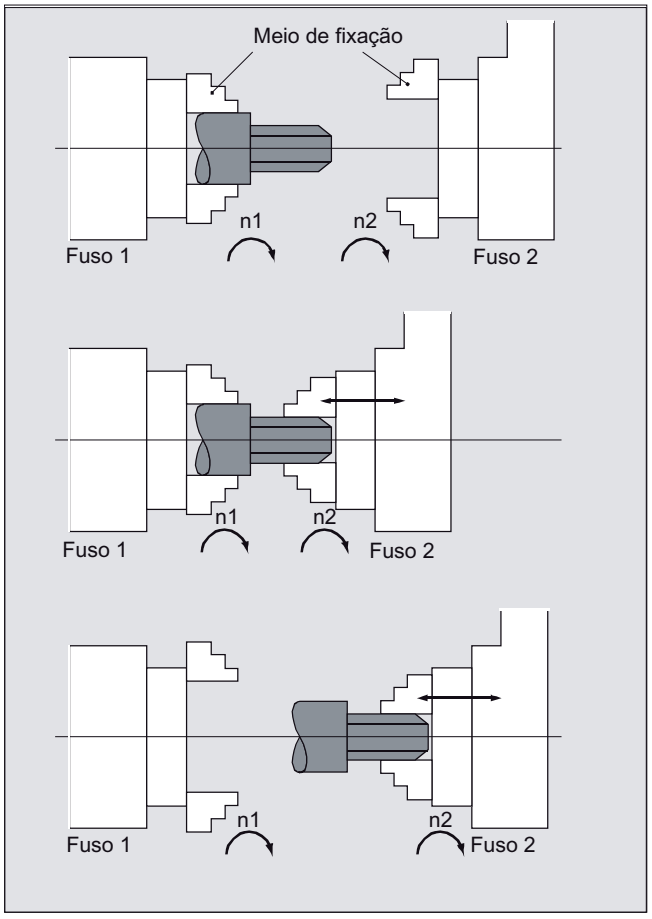
9.5.1 Fuso sincronizado: Programação (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC)

Função

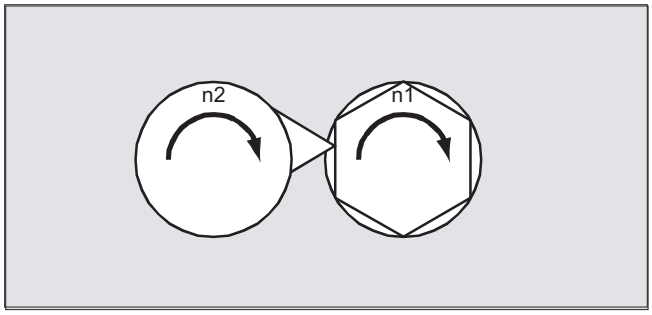
A função de fuso sincronizado permite o movimento sincronizado de dois fusos (fuso escravo FS e fuso mestre LS), por exemplo, para transferência aérea de peças de trabalho.

A função oferece os seguintes modos:

- Sincronismo de rotação ($n_{FS} = n_{LS}$)
- Sincronismo de posição ($\phi_{FS} = \phi_{LS}$)
- Sincronismo de posição com deslocamento angular ($\phi_{FS} = \phi_{LS} + \Delta\phi$)



Através da definição de uma relação de transmissão diferente de 1 entre os fusos mestre e escravo também é possível executar uma usinagem de poliedros (torneamento de polígonos).



Sintaxe

```
COUPDEF(<FS>,<LS>,<ÜFS>,<ÜLS>,<mudança de blocos>,<tipo de
acoplamento>)
COUPON(<FS>,<LS>,<POSFS>)
COUPONC(<FS>,<LS>)
COUPOF(<FS>,<LS>,<POSFS>,<POSLS>)
COUPOFS(<FS>,<LS>)
COUPOFS(<FS>,<LS>,<POSFS>)
COUPRES(<FS>,<LS>)
COUPDEL(<FS>,<LS>)
WAITC(<FS>,<mudança de blocos>,<LS>,<mudança de blocos>)
```

Indicação

Forma escrita abreviada

Para as instruções COUPOF, COUPOFS, COUPRES e COUPDEL também é possível uma forma escrita abreviada sem a indicação do fuso mestre.

Significado

COUPDEF:	Definição/alteração de acoplamento específica de usuário
COUPON:	Ativar acoplamento. A partir da atual rotação o fuso escravo sincroniza com o fuso mestre
COUPONC:	Aceitação do acoplamento na ativação com programação anterior do M3 S... ou M4 S.... Uma rotação diferencial do fuso escravo é incluída imediatamente.
COUPOF:	Desativação de acoplamento. • com mudança de blocos imediata: COUPOF(<S2>,<S1>) • Mudança somente depois de passar a(s) posição(ções) de desativação <POSFS> e <POSLS>: COUPOF(<S2>,<S1>,<POSFS>) COUPOF(<S2>,<S1>,<POSFS>,<POSLS>)

COUPOFS:	Desativação de um acoplamento com parada do eixo escravo. Mudança de blocos realizada o mais rápido possível com a mudança imediata: COUPOFS(<S2>,<S1>) Mudança de blocos apenas ao ultrapassar a posição de desativação: COUPOFS(<S2>,<S1>,<POSFS>)
COUPRES:	Resetamento do parâmetro de acoplamento no MD e no SD configurado
COUPDEL:	Deletação do acoplamento definido pelo usuário
WAITC:	Esperar a condição de sincronização (NOC são cancelados no IPO na mudança de blocos)
<FS>:	Denominação do fuso escravo

Parâmetros opcionais:

<LS>:	Denominação do fuso mestre Especificação com número de fuso p. ex. S2, S1
<ÜFS>, <ÜLS>:	Relação de transmissão entre FS e LS. <ÜFS> = numerador, <ÜLS> = denominador Pré-ajuste: <ÜFS> / <ÜLS> = 1.0 ; especificação de denominador opcional
<mudança de blocos>:	Comportamento de mudança de blocos A relação de mudança de blocos é realizada pelo(a): "NOC" imediatamente "FINE" com o alcance da "Sincronização fina" "COARSE" com o alcance da "Sincronização aproximada" "IPOSTOP" com alcance do IPOSTOP, isto é, após a sincronização de valor nominal (pré-ajuste) O comportamento de mudança de blocos está ativo modalmente.
<tipo de acoplamento>:	Tipo de acoplamento: Acoplamento entre FS e LS "DV" Acoplamento de valor nominal (pré-ajuste) AV Acoplamento de valor real "VV" Acoplamento de velocidade O tipo de acoplamento está ativo modalmente.
<POSFS>:	Deslocamento angular entre fuso mestre e fuso escravo Faixa de valores: 0°… 359,999°
<POSFS>,<POSLS>:	Posições de desativação do fuso escravo e fuso mestre "A mudança de blocos é liberada após a ultrapassagem do POS_{FS}, POS_{LS}" Faixa de valores: 0°… 359,999°

Exemplos

Exemplo 1: Trabalho com fuso mestre e fuso escravo

```
Programação                              Comentário
                                         ; Fuso de comando = fuso mestre = fuso 1
                                         ; Fuso escravo = fuso 2
N05 M3 S3000 M2=4 S2=500                 ; O fuso mestre gira com 3000 rpm,
                                           o fuso escravo gira com 500 rpm.
N10 COUPDEF(S2,S1,1,1,"NOC","Dv")        ; Definição do acoplamento (também pode ser projetada).
...
N70 SPCON                                ; Definição de fuso mestre no controle de posição
                                           (acoplamento de valor nominal).
N75 SPCON(2)                             ; Definição de fuso escravo no controle de posição.
N80 COUPON(S2,S1,45)                     ; Acoplamento flutuante na posição de deslocamento =
                                           45 graus.
...
N200 FA[S2]=100                          ; Velocidade de posicionamento = 100 graus/min
N205 SPOS[2]=IC(-90)                     ; Deslocamento de 90 graus sobrepostos no sentido negativo.
N210 WAITC(S2,"Fine")                    ; Espera pela sincronização "fina".
N212 G1 X... Y... F...                   ; Usinagem
...
N215 SPOS[2]=IC(180)                     ; Desloca 180 graus sobrepostos no sentido positivo.
N220 G4 S50                              ; Tempo de espera = 50 rotações do fuso mestre
N225 FA[S2]=0                            ; Ativação da velocidade configurada (MD).
N230 SPOS[2]=IC(-7200)                   ; 20 rotações. Deslocamento com velocidade configurada no
                                           sentido negativo
...
N350 COUPOF(S2,S1)                       ; Desacoplamento flutuante, S=S2=3000
N355 SPOSA[2]=0                          ; Parada do FS em zero grau.
N360 G0 X0 Y0
N365 WAITS(2)                            ; Espera pelo fuso 2.
N370 M5                                  ; Parada do FS.
N375 M30
```

Exemplo 2: Programação de uma rotação diferencial

```
Programação                    Comentário
                               ; Fuso de comando = fuso mestre = fuso 1
                               ; Fuso escravo = fuso 2
N01 M3 S500                    ; O fuso mestre gira com 500 rpm.
N02 M2=3 S2=300                ; O fuso escravo gira com 300 rpm.
...
N10 G4 F1                      ; Tempo de espera do fuso mestre.
N15 COUPDEF (S2,S1,-1)         ; Fator de acoplamento com relação de transmissão -1:1
N20 COUPON(S2,S1)              ; Ativar acoplamento. A rotação do fuso escravo resulta da rotação do
                                 fuso escravo e do fator de acoplamento.
...
N26 M2=3 S2=100                ; Programação de uma rotação diferencial.
```

Exemplo 3: Exemplos da transferência de um movimento para rotação diferencial

1. Ativação do acoplamento com programação anterior do fuso escravo com COUPON

```
Programação                    Comentário
                               ; Fuso de comando = fuso mestre = fuso 1
                               ; Fuso escravo = fuso 2
N05 M3 S100 M2=3 S2=200        ; O fuso mestre gira com 100 rpm, o fuso escravo gira com 200 rpm.
N10 G4 F5                      ; Tempo de espera = 5 segundos do fuso mestre
N15 COUPDEF(S2,S1,1)           ; A relação de transmissão do FS com o LS é 1,0 (pré-ajuste).
N20 COUPON(S2,S1)              ; Acoplamento flutuante no fuso mestre.
N10 G4 F5                      ; O fuso escravo gira com 100 rpm.
```

2. Ativação do acoplamento com programação anterior do fuso escravo com COUPONC

```
Programação                    Comentário
                               ; Fuso de comando = fuso mestre = fuso 1
                               ; Fuso escravo = fuso 2
N05 M3 S100 M2=3 S2=200        ; O fuso mestre gira com 100 rpm, o fuso escravo gira com 200 rpm.
N10 G4 F5                      ; Tempo de espera = 5 segundos do fuso mestre
N15 COUPDEF(S2,S1,1)           ; A relação de transmissão do FS com o LS é 1,0 (pré-ajuste).
N20 COUPONC(S2,S1)             ; Acoplar flutuante no fuso mestre e assumir a rotação anterior para
                                 S2.
N10 G4 F5                      ; O S2 gira com 100 rpm + 200 rpm = 300 rpm
```

3. Ativação do acoplamento com fuso escravo parado com COUPON

```
Programação                    Comentário
                               ; Fuso de comando = fuso mestre = fuso 1
                               ; Fuso escravo = fuso 2
N05 SPOS=10 SPOS[2]=20         ; Fuso escravo S2 em modo de posicionamento.
N15 COUPDEF(S2,S1,1)           ; A relação de transmissão do FS com o LS é 1,0 (pré-ajuste).
N20 COUPON(S2,S1)              ; Acoplamento flutuante no fuso mestre.
N10 G4 F1                      ; O acoplamento é encerrado, o S2 permanece parado em 20 graus.
```

4. Ativação do acoplamento com fuso escravo parado com COUPONC

Indicação

Modo de posicionamento ou modo de eixo

Se o fuso escravo estiver em modo de posicionamento ou em modo de eixo antes do acoplamento, então o fuso escravo se comporta de maneira igual para COUPON(<FS>,<LS>) e COUPONC(<FS>,<LS>).

ATENÇÃO

Fuso mestre e modo de eixo

Se o fuso mestre estiver em modo de eixo antes da definição do acoplamento, vigorará, mesmo após a ativação do acoplamento, o valor limite de velocidade do dado de máquina:

MD32000 $MA_MAX_AX_VELO (velocidade de eixo máxima)

Para evitar este comportamento, o eixo deve ser ativado em modo de fuso (M3 S... ou M4 S...) antes da definição do acoplamento.

Outras informações

Definição do par de fusos sincronizados

Acoplamento configurado:

Com o acoplamento configurado os fusos mestre e escravo são definidos através de dado de máquina. Os fusos configurados não podem ser alterados no programa de peça. A parametrização do acoplamento pode ser modificada no programa de peça com COUPDEF (condição: nenhuma proteção de gravação definida).

Acoplamento definido pelo usuário:

Com COUPDEF um acoplamento pode ser redefinido ou alterado no programa de peça. Se já existe um acoplamento ativo, este deve ser cancelado primeiro com COUPDEL, antes da definição de um novo acoplamento.

Definição de acoplamento: COUPDEF

Um acoplamento é totalmente definido por:

```
COUPDEF(<FS>,<LS>,<ÜFS>,<ÜLS>, comportamento de mudança de blocos, tipo de
acoplamento)
```

Fuso escravo (FS) e fuso mestre (LS)

O acoplamento é definido de forma clara com os nomes de eixo para fuso escravo (FS) e fuso mestre (LS). Os nomes de eixo devem ser programados com cada instrução `COUPDEF`. Os demais parâmetros de acoplamento estão ativos modalmente e somente precisam ser programados, quando forem alterados.

Exemplo:

```
COUPDEF(S2,S1)
```

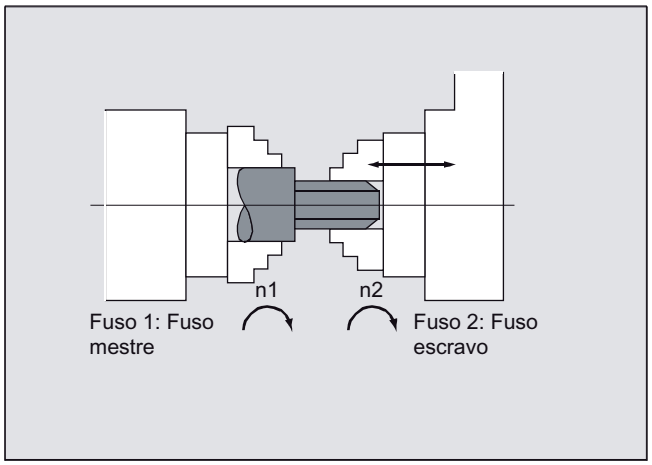
Relação de transmissão ÜFS / ÜLS

A relação de transmissão é especificada como relação de rotação entre o fuso escravo (numerador) e o fuso mestre (denominador). O numerador deve ser programado. Se nenhum denominador for programado, o denominador passa a ser = 1.0.

Exemplo:

Fuso escravo S2 e fuso mestre S1, relação de transmissão = 1 / 4 = 0.25.

```
COUPDEF(S2,S1,1.0, 4.0)
```



Indicação

A relação de transmissão também pode ser alterada com o acoplamento ativo e os fusos girando.

Comportamento de mudança de blocos NOC, FINE, COARSE, IPOSTOP

Para a programação do comportamento de mudança de blocos é possível a seguinte forma escrita abreviada:

- `"NO"`: imediatamente (pré-ajuste)
- `"FI"`: com o alcance da "Sincronização fina"
- `"CO"`: com o alcance da "Sincronização aproximada"
- `"IP"`: com alcance do IPOSTOP, isto é, após a sincronização de valor nominal

Tipo de acoplamento DV, AV

⚠ CUIDADO
O tipo de acoplamento somente pode ser modificado com o acoplamento desativado!

Ativação do modo sincronizado COUPON, POSFS

- Ativação do acoplamento com qualquer referência angular entre LS e FS:
  - `COUPON(S2,S1)`
  - `COUPON(S2,S1,<POSFS>)`
  - `COUPON(S2)`
- Ativação do acoplamento com deslocamento angular `<POSFS>`

  Para acoplamento sincronizado de posição em peças de trabalho perfiladas.
  `<POSFS>`refere-se à posição de 0° do fuso mestre no sentido de giro positivo
  Faixa de valores`<POSFS>`: 0°… 359,999°
  - `COUPON(S2,S1,30)`

  Desse modo também pode-se modificar o deslocamento angular já com o acoplamento ativo.

Posicionamento do fuso escravo

Com o acoplamento de fusos sincronizados ativado, também se pode posicionar fusos escravos dentro de ±180° independentemente do movimento realizado pelo fuso mestre.

Posicionamento SPOS

O fuso escravo pode ser interpolado com `SPOS=...`.

Exemplo:

```
SPOS[2] = IC(-90)
```

Mais informações sobre o `SPOS` estão disponíveis no(a):
Literatura:
Manual de programação - Fundamentos

Rotação diferencial M3 S... ou M4 S...

Uma rotação diferencial resulta da sobreposição (diferença de sinal) de duas origens de rotação e ela é novamente programada para o fuso escravo, p. ex. com `S<n>=...` ou `M<n>=3`, `M<n>=4` em modo de controle de rotação durante um acoplamento de fusos sincronizados. Neste caso este componente de rotação é derivado do fuso mestre através do fator de acoplamento e adicionado ao fuso escravo com o sinal correto.

Indicação

Com o sentido de rotação `M3` ou `M4` a rotação `S...` também deve ser reprogramada, pois será emitido um alarme se faltar esta programação.

Para mais informações sobre rotação diferencial, veja:
Literatura:
Manual de funções ampliadas; Fuso sincronizado (S3)

Rotação diferencial para COUPONC

Transferência de um movimento até a rotação diferencial

Através da ativação de um acoplamento de fusos sincronizados com COUPONC ocorre a sobreposição da atual rotação do fuso escravo (M3 S... ou M4 S...).

Indicação

Liberação da sobreposição

Uma sobreposição da rotação do fuso (M3 S... ou M4 S...) através do acoplamento de fusos sincronizados COUPONC somente estará ativa se a sobreposição estiver habilitada.

Restrição dinâmica do fuso mestre

A dinâmica do fuso mestre deve ser restringida de modo que no caso de uma sobreposição do fuso escravo, os valores limite de dinâmica deste último não sejam ultrapassados.

Velocidade, aceleração: FA, ACC, OVRA, VELOLIMA

A velocidade axial e aceleração axial de um fuso escravo são programáveis com:

- FA[SPI(S<n>)] e FA[S<n>] (velocidade axial), respectivamente
- ACC[SPI(S<n>)] e ACC[S<n>] (aceleração axial), respectivamente
- OVRA[SPI(S<n>)] e OVRA[S<n>] (Override axial), respectivamente
- VELOLIMA[SPI(S<n>)] e VELOLIMA[S<n>] (aceleração e redução de velocidade axial), respectivamente

Com <n> = 1, 2, 3, ... (número de fuso dos fusos escravos)

Literatura:
Manual de programação Fundamentos

Indicação

Componente de aceleração JERKLIMA[S<n>]

A programação de uma aceleração e redução de velocidade axial não está ativa nos fusos no momento.

Mais informações sobre a configuração da dinâmica axial estão disponíveis no(a):
Literatura:
Manual de funções ampliadas; Eixos rotativos (R2)

Comportamento de mudança de blocos programável WAITC

Com o WAITC define-se o comportamento de mudança de blocos, por exemplo, após a alteração de parâmetros de acoplamento ou processos de posicionamento, com diferentes condições de sincronização (aproximada, fina, IPOSTOP). Se não for especificada nenhuma condição de sincronização, é aplicado o comportamento de mudança de blocos indicado na definição COUPDEF.

Exemplo:

Espera pelo alcance (preenchimento) da condição de sincronização de acordo com o `COUPDEF`

```
WAITC( )
```

Espera pelo alcance (preenchimento) da condição de sincronização `FINE` para fuso escravo S2 e `COARSE` para fuso escravo S4:

```
WAITC(S2,"FINE",S4,"COARSE")
```

Desativação de acoplamento COUPOF

Com o `COUPOF` define-se o comportamento de desativação do acoplamento:

- Desativação do acoplamento com mudança de blocos imediata:
  - `COUPOF(S2,S1)` (com indicação do fuso mestre)
  - `COUPOF(S2)` (sem indicação do fuso mestre)
- Desativação do acoplamento depois de ultrapassar as posições de desativação. A mudança de blocos é realizada depois da ultrapassagem das posições de desativação.
  - `COUPOF(S2,S1,150)` (posição de desativação do FS: 150°)
  - `COUPOF(S2,S1,150,30)` (posição de desativação do FS: 150°, LS: 30°)

Desativação do acoplamento com parada do fuso escravo COUPOFS

Com o `COUPOFS` define-se o comportamento de desativação do acoplamento com parada do fuso escravo:

- Desativação do acoplamento com parada do fuso escravo e mudança de blocos imediata:
  - `COUPOFS(S2,S1)` (com indicação do fuso mestre)
  - `COUPOFS(S2)` (sem indicação do fuso mestre)
- Desativação do acoplamento após ultrapassar as posições de desativação com parada do fuso escravo. A mudança de blocos é realizada depois da ultrapassagem das posições de desativação.
  - `COUPOFS(S2,S1,150)` (posição de desativação do FS: 150°)

Deletação de acoplamentos COUPDEL

Com o `COUPDEL` deleta-se o acoplamento:

- `COUPDEL(S2,S1)` (com indicação do fuso mestre)
- `COUPDEL(S2)` (sem indicação do fuso mestre)

Resetamento dos parâmetros de acoplamento COUPRES

Com o `COUPRES` são ativados os valores do acoplamento parametrizados nos dados de máquina e de ajuste:

- `COUPRES(S2,S1)` (com indicação do fuso mestre)
- `COUPRES(S2)` (sem indicação do fuso mestre)

Variáveis de sistema

Atual estado de acoplamento do fuso escravo

O atual estado de acoplamento de um fuso escravo pode ser lido através da seguinte variável de sistema:

`$AA_COUP_ACT[<FS>]`

Valor	Significado
0	Nenhum acoplamento ativo
4	Acoplamento de fusos sincronizados ativo
Nota: Outros valores das variáveis de sistema referem-se ao modo de eixo **Literatura:** Manual de listas de Variáveis de sistema	

Atual deslocamento angular

O atual deslocamento angular de um fuso escravo em relação ao fuso mestre pode ser lido através da seguinte variável de sistema:

- `$AA_COUP_OFFS[<FS>]` (deslocamento angular por parte do valor nominal)
- `$VA_COUP_OFFS[<FS>]` (deslocamento angular por parte do valor real)

Indicação

Depois de anular a liberação do regulador com o acoplamento ativado e em modo de acompanhamento, ao restabelecer a liberação do regulador é ajustado outro deslocamento de posição do que o valor programado originalmente. Neste caso o deslocamento de posição alterado pode ser lido e, se necessário, corrigido no programa de peça.

9.6 Grupo mestre/escravo (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS)

Função

O acoplamento mestre/escravo antes do SW 6.4 permite o acoplamento dos eixos escravos com seu eixo mestre apenas com os eixos envolvidos em estado parado.

A ampliação do estado do SW 6.5 permite o acoplamento e separação dos fusos **rotativos** e de rotação controlada e a configuração dinâmica.

Sintaxe

`MASLON(Slv1,Slv2,..., )`	
`MASLOF(Slv1,Slv2,..., )`	
`MASLDEF(Slv1,Slv2,..., eixo mestre)`	Aplicação para configuração dinâmica
`MASLDEL(Slv1,Slv2,..., )`	Aplicação para configuração dinâmica
`MASLOFS(Slv1, Slv2, ..., )`	Ampliação para fuso escravo

Indicação

Com `MASLOF/MASLOFS` é suprimida a parada de pré-processamento implícita. Condicionadas à falta de parada de pré-processamento, as variáveis de sistema $P não enviam valores atualizados aos eixos escravos até o momento de uma nova programação.

Significado

Geral

`MASLON`	Ativar um acoplamento temporário
`MASLOF`	Separar um acoplamento ativo. Nos caso dos fusos devem ser observadas as ampliações.

Ampliação da configuração dinâmica

`MASLDEF`	Criar/alterar o acoplamento definido por usuário através de dados de máquina ou a partir do programa de peça.
`MASLOFS`	Separar o acoplamento de modo similar ao `MASLOF` e desacelerar automaticamente o fuso escravo.
`MASLDEL`	Separar o grupo de eixos mestre/escravo e deletar a definição do grupo.
`Slv1, Slv2, ...`	Eixos escravos guiados por um eixo mestre.
`Eixo mestre`	Eixo que guia os eixos escravos definidos em um grupo mestre/escravo.

Exemplos

Exemplo 1: Configuração dinâmica de um acoplamento mestre/escravo a partir do programa de peças

Configuração dinâmica de um acoplamento mestre/escravo a partir do programa de peça:

O eixo relevante após um giro do contentor de eixos deve tornar-se um eixo mestre.

```
Código de programa   Comentário
MASLDEF(AUX,S3)      ; S3 mestre para AUX
MASLON(AUX)          ; Acoplamento ativado para AUX
M3=3 S3=4000         ; Sentido de giro à direita
MASLDEL(AUX)         ; Deletação da configuração e separação do acoplamento
AXCTSWE(CT1)         ; Giro do contentor
```

Exemplos

Exemplo 2: Acoplamento de valor real de um eixo escravo

Acoplamento de valor real de um eixo escravo de mesmo valor do eixo mestre com PRESETON.

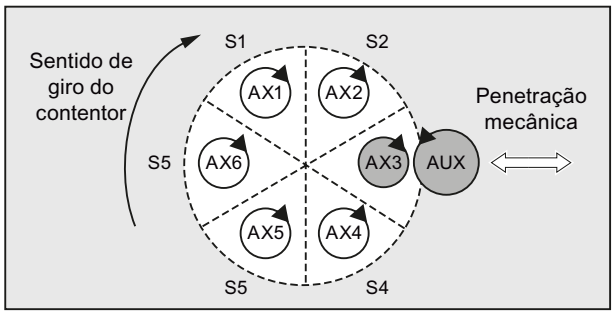
Em um acoplamento permanente de mestre e escravo deve-se modificar o valor real no eixo ESCRAVO mediante PRESETON.

```
Código de programa                                  Comentário
N37262 $MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=0         ; Desativar brevemente o
                                                      acoplamento permanente.
N37263 NEWCONF
N37264 STOPRE
MASLOF(Y1)                                          ; Acoplamento temporário
                                                      desativado.
N5 PRESETON(Y1,0,Z1,0,B1,0,C1,0,U1,0)               ; Definir o valor real dos
                                                      eixos escravos não
                                                      referenciados, pois estes
                                                      estão ativados com
                                                      Power On.
N37262 $MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=1         ; Ativar acoplamento
                                                      permanente.
N37263 NEWCONF
```

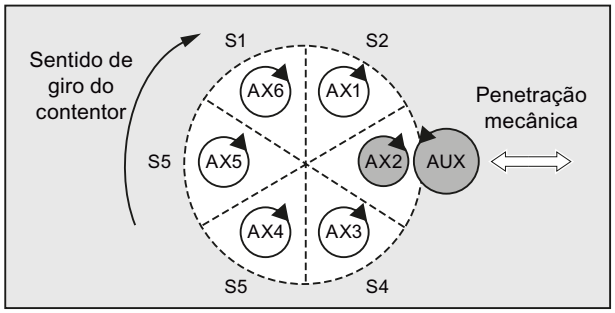
Exemplo 3: Seqüência de acoplamento da posição 3 / contentor CT1

Para permitir o restabelecimento do acoplamento com outro fuso após o giro do contentor, primeiro se deve separar o acoplamento anterior, deletada a configuração e configurado o novo acoplamento.

Situação inicial:



Após o giro em uma ranhura:



Literatura:

Manual de funções ampliadas; Vários painéis de comando e NCUs (B3), Capítulo: "Contentor de eixo"

Outras informações

Geral

MASLOF	Nos fusos em modo de controle de rotação, esta instrução é executada imediatamente. Os fusos escravos que estão em rotação neste momento mantém suas rotações até a reprogramação da rotação.

Ampliação da configuração dinâmica

MASLDEF	Definição de um grupo mestre/escravo a partir do programa de peça. Antes a definição era realizada exclusivamente através de dados de máquina.
MASLDEL	A instrução cancela a atribuição dos eixos escravos ao eixo mestre e separa simultaneamente o atual acoplamento, de forma similar ao MASLOF. As definições de mestre/escravo acordadas nos dados de máquina são conservadas.
MASLOFS	A instrução MASLOFS pode ser utilizada para desacelerar automaticamente os fusos escravos na separação do acoplamento. Nos eixos e fusos em modo de posicionamento, o acoplamento somente é encerrado e separado em estado parado.

Indicação

O valor real para o eixo escravo pode ser sincronizado com PRESETON no mesmo valor do eixo mestre. Para isso, o acoplamento permanente de mestre e escravo deve ser desativado brevemente para ajustar o valor real do eixo escravo não referenciado com o valor do eixo mestre com o Power On. Em seguida é restabelecido o acoplamento permanente.

O acoplamento mestre/escravo permanente é ativado com o ajuste de dado de máquina MD37262 $MA_MS_COUPLING_ALWAYS_ACTIVE = 1 e não influi nos comandos de linguagem do acoplamento temporário.

Comportamento de acoplamento para fusos

Para fusos que estão em modo de controle de rotação se define o comportamento de acoplamento do MASLON, MASLOF, MASLOFS e MASLDEL de forma explícita através do dado de máquina MD37263 $MA_MS_SPIND_COUPLING_MODE.

No ajuste padrão com MD37263 = 0 o acoplamento e a separação dos eixos escravos são realizados exclusivamente com os eixos participantes parados. MASLOFS corresponde ao MASLOF.

Com MD37263 = 1 executa-se imediatamente a instrução de acoplamento e também sua movimentação. O acoplamento é imediatamente encerrado com MASLON e imediatamente separado com MASLOFS ou MASLOF. Os fusos escravos que estão em rotação neste momento são desacelerados automaticamente com MASLOFS e com MASLOF conservam suas rotações até a reprogramação da rotação.

Ações síncronas de movimentos 10

10.1 Fundamentos

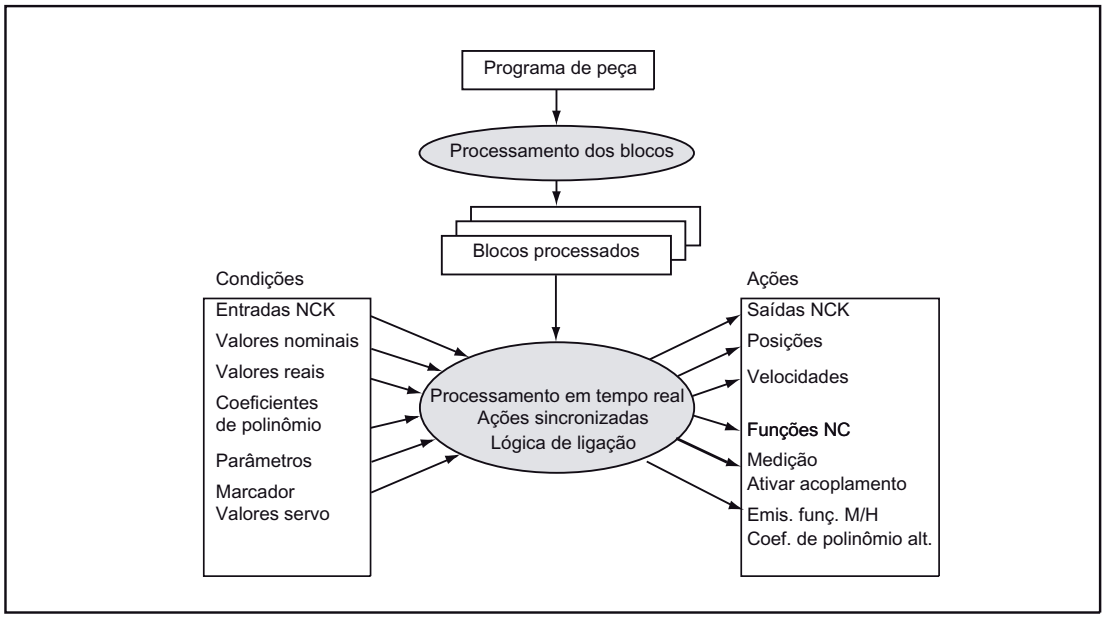
Função

As ações sincronizadas oferecem a possibilidade de executar ações sincronizadas com os blocos de usinagem.

O momento da execução das ações pode ser definido através de condições. As condições são monitoradas no ciclo de interpolação. Com isso as ações representam uma reação aos eventos de tempo real, e sua execução não está associada aos limites dos blocos.

Uma ação sincronizada também contém informações para a atividade das ações e para consultas das variáveis de processamento principal programadas e com isso a executabilidade das ações que devem ser iniciadas. Desse modo uma ação somente pode ser iniciada uma vez ou então de forma cíclica (sempre no ciclo de interpolação).

Possíveis aplicações

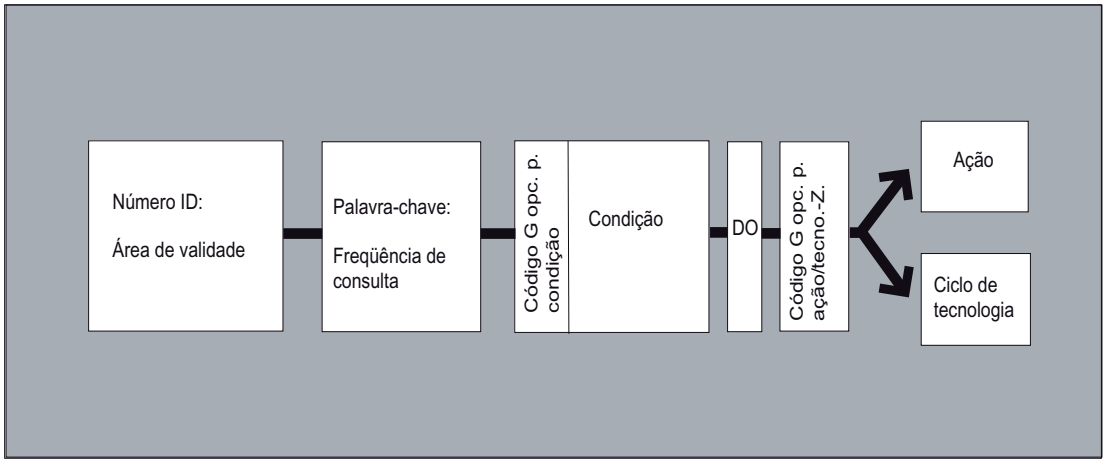


- Otimização de aplicações críticas de tempo de execução (p. ex. troca de ferramentas)
- Reação rápida à eventos externos
- Programar regulagens AC
- Ajustar funções de segurança
- ....

Programação

Uma ação sincronizada encontra-se isolada em um bloco e atua no próximo bloco executável de uma função de máquina (p. ex. o movimento de deslocamento com `G0`, `G1`, `G2`, `G3`).

As ações sincronizadas são constituídas por até 5 elementos de comando com diferentes tarefas:



Sintaxe:

```
DO <ação1> <ação2> …
<PALAVRA-CHAVE> <condição> DO <ação1> <ação2> …
ID=<n> <PALAVRA-CHAVE> <condição> DO <ação1> <ação2> ...
IDS=<n> <PALAVRA-CHAVE> <condição> DO <ação1> <ação2> …
```

Significado:

`DO`	Instrução para ativação das ações programadas Atua somente com `<condição>` preenchida (se estiver programada). → Veja " Ações "

`<ação1>`
`<ação2>`
…
	Para ações que devem ser iniciadas
	Exemplos:
	• Atribuir variáveis
	• Iniciar ciclo de tecnologia

`<PALAVRA-CHAVE>`	Através da palavra-chave (`WHEN`, `WHENEVER`, `FROM` ou `EVERY`) se define o controle cíclico da `<condição>` de uma ação sincronizada. → Veja " Controle cíclico da condição "

<condição>	Lógica de ligação para variáveis de processamento principal A condição é controlada no ciclo de interpolação.
ID=<n> e IDS=<n>	Número de identificação Com o número de identificação se define a área de validade e a posição na seqüência de usinagem. → Veja " Área de aplicação e seqüência de usinagem "

Coordenação de ações sincronizadas/ciclos de tecnologia

Para coordenação de ações sincronizadas e ciclos de tecnologia estão disponíveis os seguintes comandos:

Comando	Significado
CANCEL(<n>)	Cancelar ações sincronizadas → Veja " Cancelar ação sincronizada "
LOCK(<n>)	Bloquear ações sincronizadas
UNLOCK(<n>)	Liberar ações sincronizadas
RESET	Resetar ciclo de tecnologia Referente ao LOCK, UNLOCK e RESET: → veja " Bloquear, liberar, resetar "

Exemplo

Código de programa	Comentário
WHEN $AA_IW[Q1]>5 DO M172 H510	; Se o valor real do eixo Q1 ultrapassar 5 mm, as funções auxiliares M172 e H510 são informadas à interface PLC.

10.1.1 Área de validade e seqüência de usinagem (ID, IDS)

Função

Área de validade

A área de validade de uma ação sincronizada é definida através da identificação ID ou IDS:

Nenhum ID modal:	Ação sincronizada ativada por blocos em modo automático
ID	Ação sincronizada ativada por blocos em modo automático até o fim do programa
IDS:	Ação sincronizada estática, ativada modalmente em cada modo de operação, mesmo além do fim do programa

Aplicações

- Loops AC em modo JOG
- Lógica de ligação para Safety Integrated
- Funções de monitoração, reações à estados da máquina em todos modos de operação

Seqüência de usinagem

Ações sincronizadas ativadas modal ou estaticamente são executadas na seqüência de seus números `ID`- ou `IDS` (`ID=<n>` ou `IDS=<n>`) no ciclo de interpolação.

As ações sincronizadas ativadas relativas ao bloco (sem número ID) são executadas na seqüência programada após a execução das ações sincronizadas ativadas modalmente.

Indicação

Através dos ajustes nos dados de máquina se pode proteger as ações sincronizadas ativas contra alterações ou cancelamentos (→ Fabricante da máquina!).

Programação

Sintaxe	Significado
nenhum ID modal	Uma ação sincronizada somente está ativa em **modo automático**. Ela somente vale para o bloco executável seguinte (bloco com instrução de movimento ou outra ação de máquina), portanto, ela está ativa **por blocos**. Exemplo: `WHEN $A_IN[3]==TRUE DO $A_OUTA[4]=10`
`ID=<n> ...`	A ação sincronizada atua **modalmente** nos blocos seguintes e é desativada com `CANCEL(<n>)` ou com a programação de uma nova ação sincronizada de mesmo ID. As ações sincronizadas ativas no bloco `M30` retardam o fim do programa. As ações sincronizadas ID somente atuam em **modo automático**. Faixa de valores de **<n>**: 1 ... 255 Exemplo: `ID=2 EVERY $A_IN[1]==1 DO POS[X]=0`
`IDS=<n>`	As ações sincronizadas estáticas atuam **modalmente em todos modos de operação**. Elas também permanecem ativas além do fim do programa e podem ser ativadas diretamente após o Power On através de uma ASUP. Desse modo é possível ativar ações que devem ser executadas independentemente do modo de operação selecionado no NC. Faixa de valores de **<n>**: 1 ... 255 Exemplo: `IDS=1 EVERY $A_IN[1]==1 DO POS[X]=100`

10.1.2 Controle cíclico da condição (WHEN, WHENEVER, FROM, EVERY)

Função

Através de uma palavra-chave se define o controle cíclico da condição de uma ação sincronizada. Se não for programada nenhuma palavra-chave, as ações da ação sincronizada são executadas em cada ciclo de interpolação.

Palavras-chave

nenhuma palavra-chave	A execução da ação não está relacionada com nenhuma condição. A ação é executada ciclicamente em cada ciclo de interpolação.
`WHEN`	A condição será consultada em cada ciclo de interpolação até que seja preenchida uma vez, a ação é executada exatamente uma vez.
`WHENEVER`	A condição é controlada em cada ciclo de interpolação. A ação correspondente é executada em cada ciclo de interpolação enquanto for preenchida a condição.
`FROM`	A condição será controlada em cada ciclo de interpolação até que seja preenchida uma vez. A ação é executada enquanto a ação sincronizada estiver ativa, isto é, também se a condição não for mais preenchida.
`EVERY`	A condição é consultada em cada ciclo de interpolação. A ação sempre é iniciada uma vez se a condição for preenchida, e será novamente executada se a condição mudar do estado FALSE para TRUE.

Variáveis de processamento principal

As variáveis utilizadas são avaliadas no ciclo de interpolação. As variáveis de processo principal nas ações sincronizadas não acionam nenhuma parada de pré-processamento.

Avaliação:

Se em um programa de peça surgirem variáveis de processamento principal (p. ex. valor real, posição de uma entrada ou saída digital, etc.), o processo será parado até que o bloco anterior seja processado e os valores das variáveis de processamento principal estiverem disponíveis.

Exemplos

Exemplo 1: Nenhuma palavra-chave

Código de programa	Comentário
`DO $A_OUTA[1]=$AA_IN[X]`	`; Retorno do valor real na saída analógica.`

Exemplo 2: WHENEVER

Código de programa	Comentário
`WHENEVER $AA_IM[X] > 10.5*SIN(45) DO …`	`; Comparação com a expressão calculada no pré-processamento.`
`WHENEVER $AA_IM[X] > $AA_IM[X1] DO …`	`; Comparação com outras variáveis de processamento principal.`
`WHENEVER ($A_IN[1]==1) OR ($A_IN[3]==0) DO ...`	`; Duas comparações interligadas.`

Exemplo 3: EVERY

Código de programa	Comentário
`ID=1 EVERY $AA_IM[B]>75 DO POS[U]=IC(10) FA[U]=900`	`; Sempre que o valor real do eixo B em MCS ultrapassar o valor 75, o eixo U deve reposicionar com avanço axial em 10.`

Outras informações

Condição

A condição representa uma expressão lógica que pode ser composta livremente através de operadores booleanos. As expressões booleanas sempre devem ser especificadas entre colchetes/parênteses.

A condição é verificada no ciclo de interpolação.

Antes da condição se pode especificar um código G. Desse modo consegue-se a existência de ajustes definidos para avaliação da condição e da execução da ação e do ciclo de tecnologia, independentemente do atual estado ativo do programa de peça. O desacoplamento das ações sincronizadas do ambiente do programa é necessário, pois as ações sincronizadas são solicitadas para executar suas ações em determinado estado de saída e a qualquer momento em função das condições de ativação.

Casos de aplicação

Definição dos sistemas de medição para avaliação de condição e ação através dos códigos `G70`, `G71`, `G700`, `G710`.

Um código G especificado em uma condição vale para a avaliação da condição e para a ação, isto se na ação não foi especificado um código G próprio.

Para cada parte de condição somente pode ser programado **um código G** do grupo de códigos G.

Possíveis condições

- Comparação de variáveis de processamento principal (entradas e saídas analógicas e digitais, entre outros)
- Ligação booleana entre os resultados de comparação
- Cálculo de expressões de tempo real
- Tempo/distância do início do bloco
- Distância do fim do bloco
- Valores de medição, resultados de medição
- Valores servo
- Velocidades, estado do eixo

10.1.3 Ações (DO)

Função

Nas ações sincronizadas pode-se programar uma ou mais ações. Todas ações programadas em um bloco são iniciadas no mesmo ciclo de interpolação.

Sintaxe

```
DO <ação1> <ação2> ...
```

Significado

`DO`	Ativa uma ação ou um ciclo de tecnologia com a condição preenchida.
`<Ação>`	Ação iniciada com condição preenchida, como por exemplo a atribuição de variável, ativação de acoplamento de eixo, definição de saídas NCK, emissão de funções M, S e H, especificação de códigos G programados, ...

códigos G podem ser programados em ações sincronizadas para ações/ciclos de tecnologia. Este código G pode especificar outro código G diferente daquele especificado na condição para todas ações no bloco e ciclos de tecnologia. Se os ciclos de tecnologia estiverem na parte da ação, então o código G permanece modal para as ações que seguem até o próximo código G após a finalização do ciclo de tecnologia.

Para cada parte de ação somente pode ser programado **um código G** do grupo de códigos (`G70`, `G71`, `G700`, `G710`).

Exemplo: Ação sincronizada com duas ações

Código de programa	Comentário
`WHEN $AA_IM[Y]>=35.7 DO M135 $AC_PARAM=50`	`; Se a condição estiver preenchida, o M135 será enviado ao PLC e o override definido em 50%.`

10.2 Operadores para condições e ações

Comparações (==, <>, <, >, <=, >=)	Nas condições podem ser comparadas variáveis ou expressões parciais. O resultado sempre será o tipo de dado BOOL. São permitidos todos operadores de comparação conhecidos.
Operadores booleanos (NOT, AND, OR, XOR)	Variáveis, constantes ou comparações podem ser ligadas entre si pelos operadores booleanos conhecidos.
Operadores Bit a Bit (B_NOT, B_AND, B_OR, B_XOR)	São possíveis os operadores Bit a Bit B_NOT, B_AND, B_OR, B_XOR.
Operações matemáticas básicas (+, -, *, /, DIV, MOD)	As variáveis de processamento principal podem ser ligadas entre si ou com constantes através das operações matemáticas básicas.
Funções matemáticas (SIN, COS, TAN, ASIN, ACOS, ABS, TRUNC, ROUND, LN, EXP, ATAN2, POT, SQRT, CTAB, CTABINV).	As funções matemáticas podem ser aplicadas em variáveis do tipo de dado REAL.
Indexação	A indexação é possível com expressões de processamento principal.

Exemplo

- **Operações matemáticas básicas ligadas**

Multiplicação e divisão são aplicadas antes da soma e subtração, a colocação de colchetes em expressões é permitida. Os operadores DIV e MOD também são permitidos para o tipo de dado REAL.

Programação	Comentário
DO $AC_PARAM[3] = $A_INA[1]-$AA_IM[Z1]	; ;subtração de duas ;variáveis de processamento principal
WHENEVER $AA_IM[x2] < $AA_IM[x1]-1.9 DO $A_OUT[5] = 1	; ;subtração de uma constante de variável
DO $AC_PARAM[3] = $INA[1]-4*SIN(45.7 $P_EP[Y])*R4	;expressão constante, calculada no processamento

- **Funções matemáticas**

Programação	Comentário
DO $AC_PARAM[3] = COS($AC_PARAM[1])	; ;

- **Expressões de tempo real**

Programação	Comentário
ID=1 WHENEVER ($AA_IM[Y]>30) AND ($AA_IM[Y]<40) DO $AA_OVR[S1]=80	; Seleção de uma janela de posição
ID=67 DO $A_OUT[1]=$A_IN[2] XOR $AN_MARKER[1]	; Avaliar 2 sinais booleanos
ID=89 DO $A_OUT[4]=$A_IN[1] OR ($AA_IM[Y]>10)	; Retornar o resultado de uma comparação

- **Variável de processamento principal identificada**

Programação	Comentário
WHEN…DO $AC_PARAM[$AC_MARKER[1]] = 3	;
Não é permitido	;
$AC_PARAM[1] = $P_EP[$AC_MARKER]	;

10.3 Variáveis de processamento principal para ações síncronas

10.3.1 Variáveis de sistema

Função

Com a ajuda das variáveis de sistema se pode ler e grava os dados do NC. As variáveis de sistema são distinguidas entre variáveis de pré-processamento ou de processamento principal. As variáveis de pré-processamento sempre são executadas no momento do pré-processamento. As variáveis de processamento principal determinam seu valor em função do atual estado do processamento principal.

Denominação

O nome das variáveis de sistema na maioria das vezes inicia com o caractere $:

Variáveis de pré-processamento:

$M...	Dados de máquina
S...	Dados de ajuste, áreas de proteção
T...	Dados de gerenciamento de ferramentas
$P	Valores programados, dados de pré-processamento
C...	Variáveis de ciclo dos ciclos envoltórios ISO
O	Dados opcionais
R	Parâmetros R

Variáveis de processamento principal:

$$A...	Atuais dados de processamento principal
$$V...	Dados servo
R...	Parâmetros R

Um 2º caractere descreve a possibilidade de acesso à variável:

N...	Valor global de NCK (geralmente o valor válido)
C...	Valor específico de canal
A...	Valor específico de eixo

O 2º caractere na maioria das vezes é utilizado para variáveis de processamento principal. As variáveis de pré-processamento, como p. ex. $P_ são executadas na maioria das vezes sem o 2º caractere.

O prefixo ($ seguido de um ou dois caracteres) sempre é seguido por uma linha sublinhada e o nome de variável seguinte (na maioria das vezes escrita com nome ou abreviação em idioma inglês).

Tipos de dados

As variáveis de processamento principal podem ter os seguintes tipos de dados:

INT	Integer para valores inteiros com sinal precedente
REAL	Real para números fracionados
BOOL	Boolean (booleano) TRUE e FALSE
CHAR	Caracteres ASCII
STRING	Seqüência de caracteres com caracteres alfanuméricos
AXIS	Endereços de eixos e fusos

As variáveis de pré-processamento também podem ter os seguintes tipos de dados:

FRAME	Transformações de coordenadas

Campos de variáveis

As variáveis de sistema podem ser criadas como campos de 1 a 3 dimensões.

São suportados os seguintes tipos de dados: BOOL, CHAR, INT, REAL, STRING, AXIS

O tipo de dado do índice pode ser do tipo INT e AXIS, onde este pode ser classificado conforme desejado.

As variáveis STRING somente podem ser criadas em 2 dimensões.

Beispiele für Felddefinitionen:

```
DEF BOOL $AA_NEWVAR[x,y,2]
DEF CHAR $AC_NEWVAR[2,2,2]
DEF INT $AC_NEWVAR[2,10,3]
DEF REAL $AA_VECTOR[x,y,z]
DEF STRING $AC_NEWSTRING[3,3]
DEF AXIS $AA_NEWAX[x,3,y]
```

Indicação

A indicação de variáveis de sistema de 3 dimensões é possível com restrições, se houver uma variável BTSS além das variáveis de sistema.

10.3.2 Conversão de tipo implícita

Função

Na atribuição de valores e transferência de parâmetros podem ser atribuídas ou transferidas variáveis de diversos tipos de dados.

A conversão de tipos implícita aciona uma conversão dos tipos dos valores.

Possíveis conversões de tipos

para	REAL	INT	BOOL	CHAR	STRING	AXIS	FRAME
de							
REAL	sim	sim*	sim[1])	–	–	–	–
INT	sim	sim	sim[1])	–	–	–	–
BOOL	sim	sim	sim	–	–	–	–

Explanações

* Na conversão de tipos de REAL para INT um valor fracionado >= 0.5 é arredondado para cima, e em caso contrário, arredondado para baixo (veja a função ROUND). Na extrapolação de valores será emitido um alarme.

1) O valor <> 0 corresponde à TRUE, o valor == 0 corresponde à FALSE

Resultados

```
Conversão de tipos de REAL ou INTEGER para BOOL
Resultado BOOL = TRUE          se o valor de REAL ou INTEGER for diferente de zero
Resultado BOOL = FALSE         se o valor de REAL ou INTEGER for igual a zero
Conversão de tipo de BOOL para REAL ou INTEGER
Resultado REAL TRUE            se o valor de BOOL = TRUE (1)
Resultado INTEGER = TRUE       se o valor de BOOL = TRUE (1)
Conversão de tipo de BOOL para REAL ou INTEGER
Resultado REAL FALSE)          se o valor de BOOL = FALSE (0)
Resultado INTEGER = FALSE      se o valor de BOOL = FALSE (0)
```

Exemplos de conversões implícitas de tipos

```
Conversão de tipo de INTEGER para BOOL
$AC_MARKER[1] = 561
ID=1 WHEN $A_IN[1] == TRUE DO $A_OUT[0]=$AC_MARKER[1]

Conversão de tipo de REAL para BOOL
R401 = 100.542
WHEN $A_IN[0] == TRUE DO $A_OUT[2]=$R401

Conversão de tipo de BOOL para INTEGER
ID=1 WHEN $A_IN[2] == TRUE DO $AC_MARKER[4] = $A_OUT[1]]

Conversão de tipo de BOOL para REAL
R401 = 100.542
WHEN $A_IN[3] == TRUE DO $R10 = $A_OUT[3]
```

10.3.3 Variáveis GUD

Variáveis GUD aptas para ações síncronas

Além das variáveis de sistema específicas, nas ações síncronas também podem ser utilizadas variáveis de usuário globais e pré-definidas para elas (GUD de ações síncronas). O número de GUDs de ações síncronasdisponíveis para o usuário é parametrizada em função do tipo de dado e da forma de acesso através dos seguintes dados de máquina:

- MD18660 $MM_NUM_SYNACT_GUD_**REAL**[<x>] = <número>
- MD18661 $MM_NUM_SYNACT_GUD_**INT**[<x>] = <número>
- MD18662 $MM_NUM_SYNACT_GUD_**BOOL**[<x>] = <número>
- MD18663 $MM_NUM_SYNACT_GUD_**AXIS**[<x>] = <número>
- MD18664 $MM_NUM_SYNACT_GUD_**CHAR**[<x>] = <número>
- MD18665 $MM_NUM_SYNACT_GUD_**STRING**[<x>] = <número>

Através do índice <x> especifica-se o módulo de dados (direitos de acesso), através do valor <número> especifica-se o número de GUDs de ações síncronas do tipo de dados (REAL, INT, ...) correspondente. Em consequência disso, no respectivo módulo de dados é criada, para cada tipo de dado, uma variável de campo de 1 dimensão com o seguinte esquema de nomes: SYG_<tipo de dado><direito de acesso>[<índice>]:

Índice <x>		Tipo de dado (MD18660 ... MD18665)					
	Módulo	REAL	INT	BOOL	AXIS	CHAR	STRING
0	SGUD	SYG_RS[i]	SYG_IS[i]	SYG_BS[i]	SYG_AS[i]	SYG_CS[i]	SYG_SS[i]
1	MGUD	SYG_RM[i]	SYG_IM[i]	SYG_BM[i]	SYG_AM[i]	SYG_CM[i]	SYG_SM[i]
2	UGUD	SYG_RU[i]	SYG_IU[i]	SYG_BU[i]	SYG_AU[i]	SYG_CU[i]	SYG_SU[i]
3	GUD4	SYG_R4[i]	SYG_I4[i]	SYG_B4[i]	SYG_A4[i]	SYG_C4[i]	SYG_S4[i]
4	GUD5	SYG_R5[i]	SYG_I5[i]	SYG_B5[i]	SYG_A5[i]	SYG_C5[i]	SYG_S5[i]
5	GUD6	SYG_R6[i]	SYG_I6[i]	SYG_B6[i]	SYG_A6[i]	SYG_C6[i]	SYG_S6[i]
6	GUD7	SYG_R7[i]	SYG_I7[i]	SYG_B7[i]	SYG_A7[i]	SYG_C7[i]	SYG_S7[i]
7	GUD8	SYG_R8[i]	SYG_I8[i]	SYG_B8[i]	SYG_A8[i]	SYG_C8[i]	SYG_S8[i]
8	GUD9	SYG_R9[i]	SYG_I9[i]	SYG_B9[i]	SYG_A9[i]	SYG_C9[i]	SYG_S9[i]
com i = 0 até (<número> - 1) Módulo: _N_DEF_DIR/_N_ ... _DEF, p.ex.para SGUD ⇒ _N_DEF_DIR/_N_**SGUD**_DEF							

Propriedades

Os GUDs de ações síncronas possuem as seguintes propriedades:

- Os GUDs de ações síncronas podem ser lidas e gravadas em ações síncronas e programas de peça / ciclos
- O acesso aos GUDs de ações síncronas pode ser realizado através do BTSS
- Os GUDs de ações síncronas são exibidos na interface de operação da HMI na área de operação "Parâmetros"
- Os GUDs de ações síncronas podem ser utilizados na HMI em Wizard, na exibição de variáveis e no protocolo de variáveis
- O tamanho de campo para GUDs de ações síncronas do tipo STRING está fixado em 32 (31 caracteres + \0).
- Mesmo se nenhum arquivo de definição para dados de usuário globais (GUD) for criado manualmente, os GUDs de ações síncronas podem ser lidos no respectivo módulo GUD da HMI através dos dados de máquina.

ATENÇÃO

As variáveis de usuário (GUD, PUD, LUD) somente podem ser definidas com o mesmo nome do GUD de ações síncronas (`DEF ... SYG_xy`), se nenhum GUD de ações síncronas for parametrizado com o mesmo nome (MD18660 - MD18665) . Estes GUDs definidos pelo usuário **não** podem ser utilizados em ações síncronas.

Direitos de acesso

Os direitos de acesso definidos em um arquivo de definições GUD continuam válidos e referem-se apenas às variáveis GUD definidas neste arquivo de definição GUD.

Comportamento de eliminação

Se o conteúdo de um determinado arquivo de definição GUD for ativado novamente, então o antigo módulo de dados GUD será deletado primeiro do sistema de arquivos ativo. Neste caso, os GUDs de ações síncronas configurados também são resetados. Este procedimento também é possível através da HMI na área de operação "Serviços" > "Definir e ativar dados de usuário (GUD)" möglich.

10.3.4 Identificadores de eixo default (NO_AXIS)

Função

Variáveis ou parâmetros do tipo AXIS que não foram inicializados com um valor podem ser atribuídos com identificadores de eixos default. As variáveis de eixo indefinidas são inicializadas com este valor default.

Nomes de eixo válidos e não inicializados são identificados através de uma consulta da variável "NO_AXIS" nas ações sincronizadas. Este identificador de eixo não inicializado é atribuído através de um identificador de eixo default configurado em dado de máquina.

Fabricante da máquina

Através de dados de máquina deve ser definido e pré-definido pelo menos um identificador de eixo válido e disponível. Entretanto, também podem ser pré-definidos todos os identificadores de eixo disponíveis e válidos. Observe as instruções do fabricante da máquina.

Indicação

Agora as novas variáveis criadas recebem automaticamente o valor de nome de eixo default armazenado no dado de máquina durante a definição.

Para mais informações sobre a definição válida para um dado de máquina, veja no(a):
Literatura:
/FBSY / Manual de funções para ações sincronizadas

Sintaxe

```
PROC UP(AXIS PAR1=NO_AXIS, AXIS PAR2=NO_AXIS)
IF PAR1 <>NO_AXIS...
```

Significado

`PROC`	Definição de subrotina
`UP`	Subrotinas para identificação
`PARn`	Parâmetro n
`NO_AXIS`	Inicialização do parâmetro formal com identificador de eixo default

Exemplo: Definição de uma variável de eixo no programa principal

```
Código de programa
DEF AXIS AXVAR
UP( , AXVAR)
```

10.3.5 Marcador de ação sincronizada ($AC_MARKER[n])

Função

A variável de campo $AC_MARKER[n] pode ser lida e gravada em ações sincronizadas. Estas variáveis podem estar na memória do sistema de arquivos ativo ou passivo.

Variável de ação sincronizada: Tipo de dado INT

$AC_MARKER[n]	Marcador/contador do tipo de dado INTEGER específico de canal
$MC_MM_NUM_AC_MARKER	Dado de máquina para ajuste do número de marcadores específicos de canal para ações sincronizadas de movimentos
n	Índice de campo das variáveis 0-n

Exemplo para leitura e gravação de variáveis de marcador

```
Código de programa
WHEN ... DO $AC_MARKER[0] = 2
WHEN ... DO $AC_MARKER[0] = 3
WHENEVER $AC_MARKER[0] == 3 DO $AC_OVR=50
```

10.3.6 Parâmetro de ação sincronizada ($AC_PARAM[n])

Função

Os parâmetros de ação sincronizada $AC_PARAM[n] servem para cálculos e como memória temporária em ações sincronizadas. Estas variáveis podem estar na memória do sistema de arquivos ativo ou passivo.

Variável de ação sincronizada: Tipo de dado REAL

Os parâmetros estão presentes uma vez por canal com o mesmo nome.

`$AC_PARAM[n]`	Variável de cálculo para ações sincronizadas de movimentos (REAL)
`$MC_MM_NUM_AC_PARAM`	Dado de máquina para ajuste da quantidade de parâmetros para ações sincronizadas de movimentos até o máximo de 20000.
`n`	Índice de campo do parâmetro 0n

Exemplo para parâmetro de ação sincronizada $AC_PARAM[n]

Código de programa

```
$AC_PARAM[0]=1.5
$AC_MARKER[0]=1
ID=1 WHEN $AA_IW[X]>100 DO $AC_PARAM[1]=$AA_IW[X]
ID=2 WHEN $AA_IW[X]>100 DO $AC_MARKER[1]=$AC_MARKER[2]
```

10.3.7 Parâmetros de cálculo ($R[n])

Função

Estas variável de campo estática serve para cálculos no programa de peça e ações sincronizadas.

Sintaxe

Programação no programa de peça:

```
REAL R[n]
REAL Rn
```

Programação em ações sincronizadas:

```
REAL $R[n]
REAL $Rn
```

Parâmetros de cálculo

O uso de parâmetros de cálculo permite:

- Armazenamento de valores que devem ser mantidos além do fim do programa, NC-Reset e Power On.
- Exibição de valores armazenados na tela de parâmetros R.

Exemplos

Código de programa	Comentário
WHEN $AA_IM[X]>=40.5 DO $R10=$AA_MM[Y]	; Utilização do R10 na ação sincronizada.
G01 X500 Y70 F1000	
STOPRE	; Parada de pré-processamento
IF R10>20	; Avaliação da variável de cálculo.

Código de programa

```
SYG_AS[2]=X
SYG_IS[1]=1
WHEN $AA_IM[SGY_AS[2]]>10 DO $R3=$AA_EG_DENOM[SYG_AS[1]],SYG_AS[2]]
WHEN $AA_IM[SGY_AS[2]]>12 DO $AA_SCTRACE[SYG_AS[2]]=1
SYG_AS[1]=X
SYG_IS[0]=1
WHEN $AA_IM[SGY_AS[1]]>10 DO $R3=$$MA_POSCTRL_GAIN[SYG_IS[0]],SYG_AS[1]]
WHEN $AA_IM[SGY_AS[1]]>10 DO $R3=$$MA_POSCTRL_GAIN[SYG_AS[1]]
WHEN $AA_IM[SGY_AS[1]]>15 DO $$MA_POSCTRL_GAIN[SYG_AS[0]], SYG_AS[1]]=$R3
```

10.3.8 Ler e gravar dados de máquina NC e dados de ajuste NC

Função

A leitura e gravação de dados de máquina NC e dados de ajuste também é possível a partir de ações sincronizadas. Na leitura e gravação de elementos de campo de dados de máquina se pode ignorar a programação de um índice. Se isto ocorre no programa de peça, então na leitura se lê o **primeiro** elemento de campo e na gravação são gravados todos elementos do campo com o valor.

Neste caso, em ações sincronizadas apenas é lido ou gravado o **primeiro** elemento.

Definição

MD, SD com

`$`: Leitura do valor no momento da interpretação das ações sincronizadas

`$$`: Leitura do valor no processamento principal

Ler valores MD e SD no momento do pré-processamento

Eles são endereçados a partir da ação sincronizada com o caractere $ e avaliados no momento do pré-processamento.

```
ID=2 WHENEVER $AA_IM[z]<$SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
Aqui é consultada a área de reversão 2 adotada como inalterada para a oscilação
```

Ler valores MD e SD no momento do processamento principal

Eles são endereçados a partir da ação sincronizada com os caracteres $$ e avaliados no momento do processamento principal.

```
ID=1 WHENEVER $AA_IM[z]<$$SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
;Aqui se adota que a posição de reversão poderia ser alterada com a operação durante
a usinagem.
```

Gravar MD e SD no momento do processamento principal

O atual direito de acesso ajustado deve permitir o acesso de gravação. A ativação para todos MD e SD está especificada na **Literatura:** /LIS/, Listas (Livro 1).

O `MD` e `SD` a ser gravado deve ser endereçado iniciado com `$$` .

Exemplo

Código de programa	**Comentário**
`ID=1 WHEN $AA_IW[X]>10 DO`	`;` Alteração da posição de comutação
`$$SN_SW_CAM_PLUS_POS_TAB_1[0]=20`	de cames de SW. Nota: As posições de comutações devem ser alteradas de 2-3 ciclos de interpolação antes de alcançar a posição.
`$$SN_SW_CAM_MINUS_POS_TAB_1[0]=30`	

10.3.9 Variável de temporizador ($AC_TIMER[n])

Função

A variável de sistema $AC_TIMER[n] permite o início de ações de acordo com tempos de espera definidos.

Variável de temporizador: Tipo de dado REAL

`$AC_TIMER[n]`	Temporizador específico de canal do tipo de dado REAL
`s`	Unidade em segundos
`n`	Índice da variável de temporizador

Ajustar o temporizador

A contagem progressiva de uma variável de temporizador é iniciada pela atribuição de valor:
`$AC_TIMER[n] = value`

n:	Número das variáveis de tempo
value:	Valor de partida (normalmente "0")

Parar temporizador

A contagem progressiva de uma variável de temporizador é parada com a atribuição de um valor negativo:
`$AC_TIMER[n]=-1`

Ler temporizador

O tempo atual pode ser lido com a variável de temporizador em atividade ou parada. Depois de parar a variável de temporizador com a atribuição de -1, o atual valor de tempo permanece parado e, mesmo assim, ainda pode ser lido.

Exemplo

Saída de um valor real através de saída analógica de 500 ms depois da identificação de uma entrada digital:

```
Código de programa                                                    Comentário
WHEN $A_IN[1]==1 DO $AC_TIMER[1]=0                                    ; Resetar e iniciar
WHEN $AC_TIMER[1]>=0.5 DO $A_OUTA[3]=$AA_IM[X] $AC_TIMER[1]=-1          temporizador
```

10.3.10 Variáveis FIFO ($AC_FIFO1[n] ... $AC_FIFO10[n])

Função

Para o armazenamento das seqüências de dados associadas existem 10 variáveis FIFO (memória de circulação) à disposição.
Tipo de dado: REAL

Aplicação:

- medição cíclica
- Execução de passagem

Em cada elemento pode ser feito o acesso de leitura e de gravação.

Variável FIFO

O número de variáveis FIFO disponíveis é definida no dado de máquina MD28260 $MC_NUM_AC_FIFO.

O número de valores que podem ser gravados em uma variável FIFO é definido através do dado de máquina MD28264 $MC_LEN_AC_FIFO. Todas variáveis FIFO possuem o mesmo tamanho.

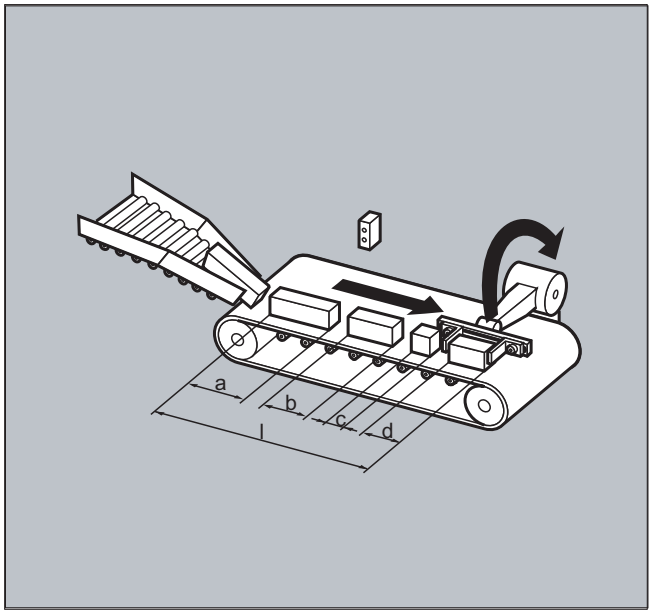
A soma de todos elementos FIFO apenas é formada se no MD28266 $MC_MODE_AC_FIFO estiver definido Bit0.

Os índices **0 até 5** tem significado especial:

Índice	Significado	
0	Na gravação:	O novo valor é armazenado no FIFO.
	Na leitura:	O elemento mais antigo é lido e removido do FIFO.
1	Acesso ao elemento mais antigo armazenado	
2	Acesso ao elemento mais novo armazenado	
3	Soma de todos elementos FIFO	
4	Número de elementos disponíveis no FIFO Cada elemento do FIFO pode ser acessado por leitura e gravação. O Reset das variáveis FIFO é realizado resetando-se o número de elementos, p. ex. para a primeira variável FIFO: $AC_FIFO1[4] = 0	
5	Atual índice de gravação relativo ao início do FIFO	
6 até $n_{máx}$	Acesso ao enésimo elemento FIFO	

Exemplo: Memória de circulação

Durante uma produção é utilizado um transportador que transporta os produtos de diferentes comprimentos (a, b, c, d). No transportador com o comprimento de transporte são transportadas diferentes quantidades de produtos em função do comprimento dos mesmos. Com uma velocidade de transporte constante deve ser realizada uma adaptação da retirada de produto do transportador através da variável de tempos de chegada dos produtos.



Código de programa	**Comentário**
`DEF REAL INTRM=2.5`	`; Distância constante entre os produtos depositados.`
`DEF REAL TOTAL=270`	`; Distância entre a posição de medição longitudinal e a posição de retirada.`
`EVERY $A_IN[1]==1 DO $AC_FIFO1[4]=0`	`; Para iniciar o processo, se deve resetar o FIFO.`
`EVERY $A_IN[2]==1 DO $AC_TIMER[0]=0`	`; Se um produto romper a barreira de luz, inicia-se a medição do tempo.`
`EVERY $A_IN[2]==0 DO $AC_FIFO1[0]=$AC_TIMER[0]*$AA_VACTM[B]`	`; Assim que a barreira de luz for liberada, calcula-se o comprimento do produto com os dados tempo e velocidade de transporte para depois ser armazenado no FIFO.`
`EVERY $AC_FIFO1[3]+$AC_FIFO1[4]*ZWI>=GESAMT DO POS[Y]=-30 $R1=$AC_FIFO1[0]`	`; Assim que a soma de todos comprimentos de produto e distâncias intermediárias forem maior ou igual ao comprimento entre a posição de deposição e a posição de retirada, é retirado o produto da posição de retirada no transportador, o respectivo comprimento de produto é removido do FIFO.`

10.3.11 Informação sobre os tipos de bloco no interpolador ($AC_BLOCKTYPE, $AC_BLOCKTYPEINFO, $AC_SPLITBLOCK)

Função

Para as ações sincronizadas estão disponíveis as seguintes variáveis de sistema, para se obter informações de um bloco que está no processamento principal:

- $AC_BLOCKTYPE
- $AC_BLOCKTYPEINFO
- $AC_SPLITBLOCK

Variáveis Blocktype e Blocktypeinfo

$AC_BLOCKTYPE		$AC_BLOCKTYPEINFO				
Valor:		Valor:				
0	diferente de 0	T	H	Z	E	Significado:
Bloco original	Bloco intermediário					Ativador para bloco intermediário:
	1	1	0	0	0	Bloco gerado internamente, mais nenhuma informação
	2	2	0	0	1	Chanfros/arredondamentos: Reta
	2	2	0	0	2	Chanfros/arredondamentos: Círculo
	3	3	0	0	1	WAB: Aproximação em linha reta
	3	3	0	0	2	WAB: Aproximação em quadrante
	3	3	0	0	3	WAB: Aproximação em semicírculo
						Correção de ferramenta:
	4	4	0	0	1	Bloco de aproximação após STOPRE
	4	4	0	0	2	Blocos de ligação no ponto de intersecção não encontrado
	4	4	0	0	3	Círculo pontilhado em cantos internos (somente para TRACYL)
	4	4	0	0	4	Círculo de desvio (ou secção de cone) em cantos externos
	4	4	0	0	5	Blocos de aproximação na supressão de correção
	4	4	0	0	6	Blocos de aproximação na reativação da correção do raio da ferramenta
	4	4	0	0	7	Divisão de blocos em função da curvatura excessiva
	4	4	0	0	8	Blocos de compensação para fresamento de topo 3D (ferramenta vetor || vetor superficial)

$AC_BLOCKTYPE		$AC_BLOCKTYPEINFO				
Valor:		Valor:				
0	diferente de 0	T	H	Z	E	Significado:
Bloco original	Bloco intermediário					Ativador para bloco intermediário:
						Suavização com:
	5	5	0	0	1	G641
	5	5	0	0	2	G642
	5	5	0	0	3	G643
	5	5	0	0	4	G644
						Bloco TLIFT com:
	6	6	0	0	1	Movimento linear do eixo tangencial e sem movimento de retração
	6	6	0	0	2	Movimento não linear do eixo tangencial (polinômio) e sem movimento de retração
	6	6	0	0	3	Movimento de retração, movimento de eixo tangencial e movimento de retração começam simultaneamente
	6	6	0	0	4	Movimento de retração, eixo tangencial inicia apenas quando uma determinada posição de retração for alcançada.
						Segmentação de curso:
	7	7	0	0	1	segmentação de curso programada sem o puncionamento ou a estampagem estar ativa
	7	7	0	0	2	segmentação de curso programada com puncionamento ou a estampagem ativa
	7	7	0	0	3	segmentação de curso gerada interna e automaticamente
						Ciclos de compilação:
	8	Aplicação ID				ID da aplicação de ciclos de compilação gerada pelo bloco

T: Posição da milhar
H: Posição da centena
Z: Posição da dezena
E: Posição da unidade

Indicação

Sempre o $AC_BLOCKTYPEINFO também contém na posição da milhar (T) o valor do tipo de bloco para o caso de existir um bloco intermediário. No $AC_BLOCKTYPE diferente de 0 não se aceita a posição da milhar.

$AC_SPLITBLOCK	
Valor:	**Significado:**
0	Bloco programado inalterável, (o bloco gerado pelo compressor também é tratado como bloco programado)
1	Existe um bloco gerado internamente ou um bloco original encurtado
3	O último bloco está em uma sequência de blocos gerada internamente ou blocos originais encurtados

Exemplo: Contagem de blocos de suavização

```
Código de programa    Comentário
$AC_MARKER[0]=0
$AC_MARKER[1]=0
$AC_MARKER[2]=0
...
                      ; Definição de ações sincronizadas com as quais
                        os blocos de suavização são contados.
                      ; Contar todos blocos de suavização no $AC_MARKER[0]:
ID=1 WHENEVER ($AC_TIMEC==0) AND ($AC_BLOCKTYPE==5) DO $AC_MARKER[0]=$AC_MARKER[0]+1
...
                      ; Contar blocos de suavização gerados com G641 no $AC_MARKER[1]:
ID=2 WHENEVER ($AC_TIMEC==0) AND ($AC_BLOCKTYPEINFO==5001) DO $AC_MARKER[1]=$AC_MARKER[1]+1
                      ; Contar blocos de suavização gerados com G642 no $AC_MARKER[2]:
ID=3 WHENEVER ($AC_TIMEC==0) AND ($AC_BLOCKTYPEINFO==5002) DO $AC_MARKER[2]=$AC_MARKER[2]+1
...
```

10.4 Ações em ações síncronas

10.4.1 Vista geral das possíveis ações em ações sincronizadas

As ações nas ações sincronizadas são constituídas por atribuições de valores, chamadas de função ou de parâmetros, palavras-chave ou ciclos de tecnologia. Através de operadores é possível realizar execuções complexas.

As possíveis aplicações são:

- Cálculos de expressões complexas no ciclo de interpolação
- Movimentos de eixo e controles de fuso
- Alterar e avaliar Online os dados de ajuste das ações sincronizadas (p. ex. retornar as posições e tempos de cames de software ao PLC ou ao módulo NC)
- Emissão de funções auxiliares ao PLC
- Ajustar funções de segurança adicionais
- Ajustar movimento sobreposto, correção Online de ferramenta e controle de distância
- Executar ações em todos modos de operação
- Ações sincronizadas controladas a partir do PLC
- Executar ciclos de tecnologia
- Emissão de sinais digitais e sinais analógicos
- Determinação de performance das ações sincronizadas no ciclo de interpolação e do tempo de processamento do controlador de posição para uma avaliação de carga/uso
- Opções de diagnóstico na interface de operação

Ação sincronizada	Descrição
DO $V…= DO $A...=	Atribuir (valores servo) Atribuir variáveis (variáveis de processamento principal)
DO $AC…[n]= DO $AC_MARKER[n]= DO $AC_PARAM[n]=	Variável especial de processamento principal Gravar ou ler marcadores de ação sincronizada Gravar ou ler parâmetros de ação sincronizada
DO $R[n]=	Gravar ou ler variáveis de cálculo
DO $MD...= DO $$SD...=	Leitura do valor do MD no momento da interpolação Gravação do valor do SD no processamento principal
DO $AC_TIMER[n]=valor inicial	Timer
DO $AC_FIFO1[n] …FIFO10[n]=	Variável FIFO
DO $AC_BLOCKTYPE= DO $AC_BLOCKTYPEINFO= DO $AC_SPLITBLOCK=	Interpretar o bloco atual (variável de processamento principal)
DO M, S e H p. ex. M07	Saída de funções auxiliares M, S e H.
DO RDISABLE	Definir bloqueio de leitura
DO STOPREOF	Cancelar parada de pré-processamento
DO DELDTG	Anulação rápida de curso restante sem parada de pré-processamento
FTCDEF(polinom., LL, UL , coefic.) DO SYNFCT(polinom., Output, Input)	Definição de polinômios Ativação de funções sincronizadas: Controle AC
DO FTOC	Correção Online de ferramenta
DO G70/G71/G700/G710	Definir sistema de medição para tarefas de posicionamento (dimensões em sistema em polegadas ou métrico)
DO POS[eixo]= / DO MOV[eixo]= DO SPOS[fuso]=	Iniciar/posicionar/parar eixos de comando Iniciar/posicionar/parar fusos
DO MOV[eixo]=valor	Iniciar/parar movimentos infinitos de um eixo de comando
DO POS[eixo]= FA [eixo]=	Avanço axial FA
ID=1 ... DO POS[eixo]= FA [eixo]= ID=2 ... DO POS[eixo]= $AA_IM[eixo] FA [eixo]=	Posicionamento a partir de ações sincronizadas
DO PRESETON(eixo, valor)	Definição de valor real (Preset a partir de ações sincronizadas)
ID=1 EVERY $A_IN[1]=1 DO M3 S… ID=2 EVERY $A_IN[2]=1 DO SPOS=	Iniciar/posicionar/parar fusos
DO TRAILON(FA,LA,fator de acoplamento) DO LEADON(FA,LA,NRCTAB,OVW)	Ativar movimento acoplado Ativar acoplamento de valor mestre
DO MEAWA(eixo)= DO MEAC(eixo)=	Ativar medição axial Ativar medição contínua
DO [campo n, m]=SET(valor, valor, ...) DO [campo n, m]=REP(valor, valor, ...)	Inicialização de variáveis de campo com listas de valores Inicialização de variáveis de campo com os mesmos valores

Ação sincronizada	Descrição
DO SETM(marcador nº)	Definir marcadores de valor
DO CLEARM(marcador nº)	Deletar marcadores de valor
DO SETAL(alarme nº)	Definir alarme de ciclo (função de segurança adicional)
DO FXS[eixo]=	Selecionar deslocamento até o encosto fixo
DO FXST[eixo]=	Alterar torque de fixação
DO FXSW[eixo]=	Alterar janela de monitoração
DO FOCON[eixo]=	Ativar deslocamento (modal) com torque/força limitada FOC
DO FOCOF[eixo]=	Desativar deslocamento com torque/força limitada
	(a ação sincronizada atua relativa ao bloco)
ID=2 EVERY $AC_BLOCKTYPE==0 DO $R1=$AC_TANEB	O ângulo entre a tangente de percurso no ponto final do atual bloco e a tangente de percurso no ponto de partida do bloco seguinte programado
DO $AA_OVR=	Override por eixo
DO $AC_OVR=	Override (correção) de trajetória
DO $AA_PLC_OVR	Override por eixo especificado pelo PLC
DO $AC_PLC_OVR	Override de percurso especificado pelo PLC
DO $AA_TOTAL_OVR	Override por eixo resultante
DO $AC_TOTAL_OVR	Override de percurso resultante
$AN_IPO_ACT_LOAD=	atual tempo de processamento da interpolação
$AN_IPO_MAX_LOAD=	tempo mais longo de processamento da interpolação
$AN_IPO_MIN_LOAD=	tempo mais curto de processamento da interpolação
$AN_IPO_LOAD_PERCENT=	atual tempo de processamento de interpolação em relação ao ciclo de interpolação
$AN_SYNC_ACT_LOAD=	atual tempo de processamento para ações sincronizadas através de todos canais
$AN_SYNC_MAX_LOAD=	tempo mais longo de processamento para ações sincronizadas através de todos canais
$AN_SYNC_TO_IPO=	porcentagem da ação sincronizada inteira
DO TECCYCLE	Executar ciclo de tecnologia
DO LOCK(n, n, ...)	Bloquear
DO UNLOCK(n, n, ...)	Liberar
DO RESET(n, n, ...)	RESET de um ciclo de tecnologia
CANCEL(n, n, ...)	Deletar as ações sincronizadas modais com a identificação ID(S) no programa de peça

10.4.2 Emissão de funções auxiliares

Função

Momento de emissão

A emissão de funções auxiliares é realizada na ação sincronizada imediatamente no momento de emissão da ação. O momento de emissão definido no dado de máquina para funções auxiliares não está ativo.

O momento de saída resulta quando a condição for preenchida.

Exemplo:

Ligar o líquido refrigerante em determinada posição de eixo:
```
WHEN $AA_IM[X]>=15 DO M07 POS[X]=20 FA[X]=250
```

Palavras-chave permitidas em ações síncronas ativas por blocos (sem ID modal)

As funções auxiliares em ações síncronas ativas por bloco (sem ID modal) somente podem ser programadas com as palavras-chave `WHEN` ou `EVERY`.

Indicação

As seguintes funções auxiliares não são permitidas a partir de ações sincronizadas:

- `M0`, `M1`, `M2`, `M17`, `M30`: Parada/fim de programa (`M2`, `M17`, `M30` possível para ciclo de tecnologia)
- `M6` ou funções M ajustadas através de dado de máquina para troca de ferramentas

---

Exemplo

Código de programa	Comentário
`WHEN $AA_IW[Q1]>5 DO M172 H510`	`; Se o valor real do eixo Q1 ultrapassar 5 mm, são informadas as funções auxiliares M172 e H510 para o PLC.`

10.4.3 Definir bloqueio de leitura (RDISABLE)

Função

Com o RDISABLE a continuação da execução do bloco no programa principal é parada quando a condição for preenchida. As ações sincronizadas de movimentos continuam sendo processadas, os blocos seguintes continuam sendo processados.

Em modo de controle da trajetória, no início do bloco com RDISABLE em ações sincronizadas sempre é ativada parada exata, independentemente se o RDISABLE se tornar ativo ou não.

Exemplo

Independentemente das entradas externas, iniciar o programa no ciclo de interpolação.

```
Código de programa                                  Comentário
...
WHENEVER $A_INA[2]<7000 DO RDISABLE                 ;  Se na entrada 2 a tensão de 7V
                                                       ficar abaixo do limite inferior,
                                                       parar a continuação ;do programa
                                                       (1000= 1V).
N10 G1 X10                                          ;  quando a condição estiver
                                                       preenchida, o bloqueio
                                                       de leitura atua no fim do N10
N20 G1 X10 Y20
...
```

10.4.4 Cancelar a parada de pré-processamento (STOPREOF)

Função

Para parada de pré-processamento STOPRE programada explicitamente ou uma parada de pré-processamento ativada implicitamente por uma ação sincronizada, o STOPREOF cancela a parada de pré-processamento após o próximo bloco de usinagem assim que a condição for preenchida.

Indicação

O STOPREOF deve ser programado com a palavra-chave `WHEN` e por blocos (sem número ID).

Exemplo

Bifurcação rápida de programa no fim do bloco.

Código de programa	Comentário
`WHEN $AC_DTEB<5 DO STOPREOF`	`; Se a distância até o fim do bloco for menor que 5 mm, cancelar ;parada de pré-processamento.`
`G01 X100`	`; Após a execução da interpolação linear é cancelada a parada de pré-processamento.`
`IF $A_INA[7]>500 GOTOF MARKE1=X100`	`; Se na entrada 7 a tensão de 5V ;for ultrapassada, saltar para o marcador 1.`

10.4.5 Anular curso restante (DELDTG)

Função

Em função de uma condição a anulação de curso restante pode ser ativada para o percurso e os eixos especificados.

Estão disponíveis:

- Anulação de curso restante rápida e preparada
- Anulação de curso restante sem parada de pré-processamento

A anulação preparada de curso restante com DELDTG permite uma reação bastante rápida ao evento de ativação e por isso que é utilizada em aplicações críticas de tempo, p. ex. se

- o tempo entre a anulação de curso restante e a partida do bloco subseqüente for muito curto.
- a condição para anulação de curso restante for preenchida com uma probabilidade muito alta.

Indicação

A denominação de eixo contida em colchetes no DELDTG somente é válida para um **um** eixo de posicionamento.

Sintaxe

Anulação do curso restante para o percurso
```
DO DELDTG
```
Anulação axial de curso restante
```
DO DELDTG(eixo1) DELDTG(eixo2) ...
```

Exemplo de anulação rápida do curso restante do percurso

```
Código de programa                                  Comentário
WHEN $A_IN[1]==1 DO DELDTG
N100 G01 X100 Y100 F1000                            ;  Se for definida a entrada, o
                                                       movimento será
                                                       cancelado
N110 G01 X…
IF $AA_DELT>50…
```

Exemplo de anulação axial rápida de curso restante

```
Código de programa                                  Comentário
Cancelamento de um movimento de
posicionamento:
ID=1 WHEN $A_IN[1]==1 DO MOV[V]=3 FA[V]=700     ;  Partida de eixo
WHEN $A_IN[2]==1 DO DELDTG(V)                   ;  Anula o curso restante, a parada
                                                   do eixo é realizada com MOV=0
Anula o curso restante em função da tensão de
entrada:
WHEN $A_INA[5]>8000 DO DELDTG(X1)               ;  Assim que na entrada 5 for
                                                   ultrapassada a tensão de 8V,
                                                   anula o curso restante
                                                   do eixo X1. O movimento de
                                                   percurso continua.
POS[X1]=100 FA[X1]=10 G1 Z100 F1000
```

Outras informações

No final do bloco de movimento, onde foi ativada a anulação preparada de curso restante, será ativada implicitamente a parada de pré-processamento.

Com isso o modo de controle da trajetória e os movimentos de eixo de posicionamento são cancelados e parados no fim do bloco com a anulação rápida de curso restante.

Indicação

Anulação preparada de curso restante

- não pode ser empregado com a correção ativa do raio da ferramenta.
- a ação somente pode ser programada em ações sincronizadas ativas relativas ao bloco (sem número ID).

10.4.6 Definição de polinômio (FCTDEF)

Função

Com FCTDEF se pode definir polinômios de 3º grau no formato $y=a_0+a_1x+a_2x^2+a_3x^3$. Estes polinômios são utilizados pela correção de ferramenta Online FTOC e pela função de avaliação SYNFCT.

Sintaxe

```
FCTDEF(polinômio n°,LLIMIT,ULIMIT,a0,a1,a2,a3)
```

Significado

`Polinômio n°`	Número do polinômio de 3ª ordem
`LLIMIT`	limite inferior para valor de função
`ULIMIT`	limite superior para valor de função
`a0, a1, a2, a3`	Coeficientes de polinômio

Estes valores também podem ser acessados através de variáveis de sistema

`$AC_FCTLL[n]`	limite inferior para valor de função
`$AC_FCTUL[n]`	limite superior para valor de função
`$AC_FCT0[n]`	a_0
`$AC_FCT1[n]`	a_1
`$AC_FCT2[n]`	a_2
`$AC_FCT3[n]`	a_3

Indicação

Gravação de variáveis de sistema

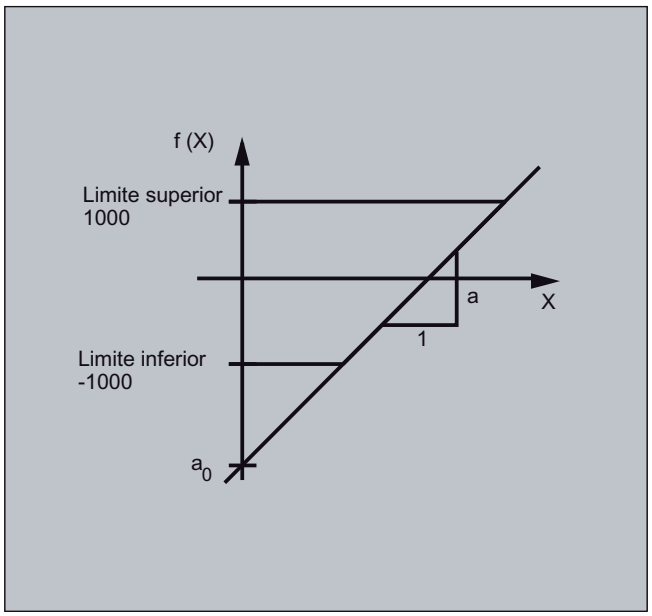
- As variáveis de sistema podem ser gravadas a partir do programa de peça ou a partir de uma ação sincronizada. Ao gravar do programa de peça deve-se providenciar a gravação sincronizada com o bloco através da programação do STOPRE.
- As variáveis de sistema `$AC_FCTLL[n]`, `$AC_FCTUL[n]`, `$AC_FCT0[n]` até `$AC_FCTn[n]` podem ser modificadas a partir das ações sincronizadas

Ao gravar a partir de ações sincronizadas os coeficientes do polinômio e os limites de valor de função estarão imediatamente ativos.

Exemplo de polinômio para segmento de reta

Com limite superior 1000, limite inferior -1000, seção de ordenada `a0=$AA_IM[X]` e a gradiente linear 1, a definição do polinômio será:

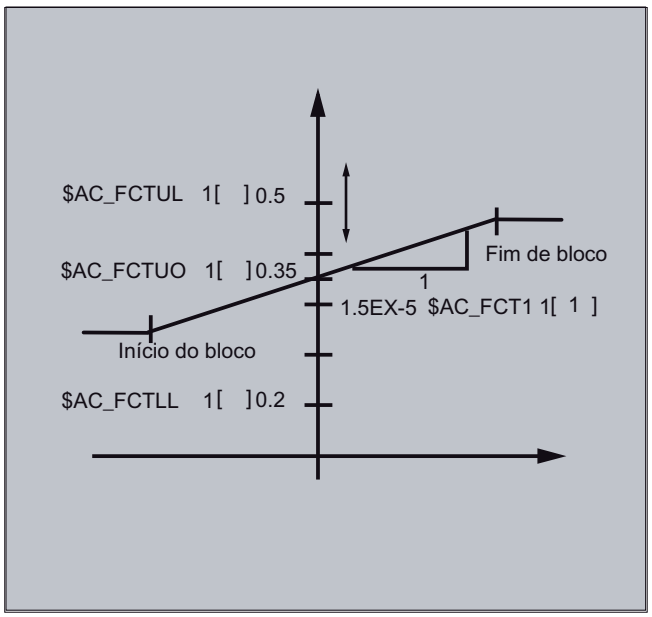
```
FCTDEF(1, -1000,1000,$AA_IM[X],1)
```



Exemplo de controle de potência do laser

Uma das possíveis aplicações da definição de polinômio é o controle de potência do laser.

Controle de potência do laser significa:
Controle de uma saída analógica em função, por exemplo, da velocidade de percurso.



Código de programa	Comentário
`$AC_FCTLL[1]=0.2`	`;` Definição dos coeficientes de polinômio
`$AC_FCTUL[1]=0.5`	
`$AC_FCT0[1]=0.35`	
`$AC_FCT1[1]=1.5EX-5`	
`STOPRE`	
`ID=1 DO $AC_FCTUL[1]=$A_INA[2]*0.1 +0.35`	`;` Altera Online o limite superior.
`ID=2 DO SYNFCT(1,$A_OUTA[1],$AC_VACTW)`	`;` Em função da velocidade de percurso (definida no $AC_VACTW) se controla a potência do laser através da saída analógica 1

Indicação

A utilização do polinômio acima citado é realizada com SYNFCT.

10.4.7 Função sincronizada (SYNFCT)

Função

O SYNFCT calcula o valor de saída de um polinômio de 3º grau definido com as variáveis de entrada. O resultado está nas variáveis de saída e pode ser, em seguida, limitado para cima e para baixo.

A função de avaliação encontra aplicação

- para controle AC (Adaptive Control),
- para controle de potência do laser
- para intercalação de posição.

Sintaxe

```
SYNFCT (polinômio n°, saída de variável de processamento principal, entrada de
variável de processamento principal)
```

Significado

Como variável de saída podem ser escolhidas variáveis que entram

- com controle aditivo
- com controle multiplicativo
- como deslocamento de posição
- diretamente

no processo de usinagem.

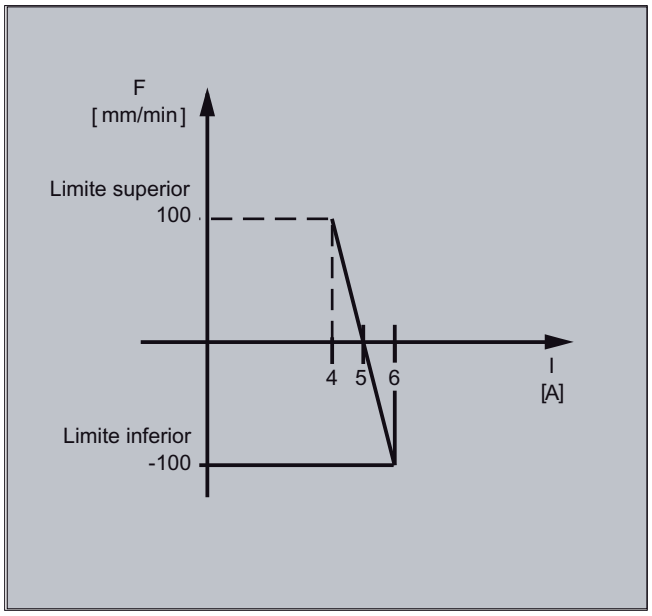
`DO SYNFCT`	Ativação da função de avaliação
`Polinômio n°`	Polinômio definido com FCTDEF (veja o subcapítulo "Definição de polinômios")
`Saída de variável de processamento principal`	Gravar variável de processamento principal
`Entrada de variável de processamento principal`	Ler variável de processamento principal

Exemplo de controle AC (aditivo)

Controle aditivo do avanço programado

Um avanço programado deve ser regulado de forma aditiva através da corrente do eixo X (eixo de aproximação):

O avanço deve variar em +/- 100 mm/min, onde a corrente oscila entre +/-1A em torno do ponto de trabalho a 5A.



1. Definição de polinômio

Definição dos coeficientes

$y = f(x) = a_0 + a_1x + a_2x^2 + a_3x^3$

a_1 = -100mm/1 min A

a_0 = -(-100)*5 =500

$a_2 = a_3 = 0$ (nenhum elemento ao quadrado e ao cubo)

Limite superior = 100

Limite inferior = -100

Disso resulta:

```
FCTDEF(1,-100,100,500,-100,0,0)
```

2. Ativar o controle AC

```
ID=1 DO SYNFCT(1,$AC_VC,$AA_LOAD[x])
```

Através do `$AA_LOAD[x]` ler a atual carga de eixo (% máx. da corrente de acionamento) ;calcular com o polinômio definido da correção do avanço de percurso.

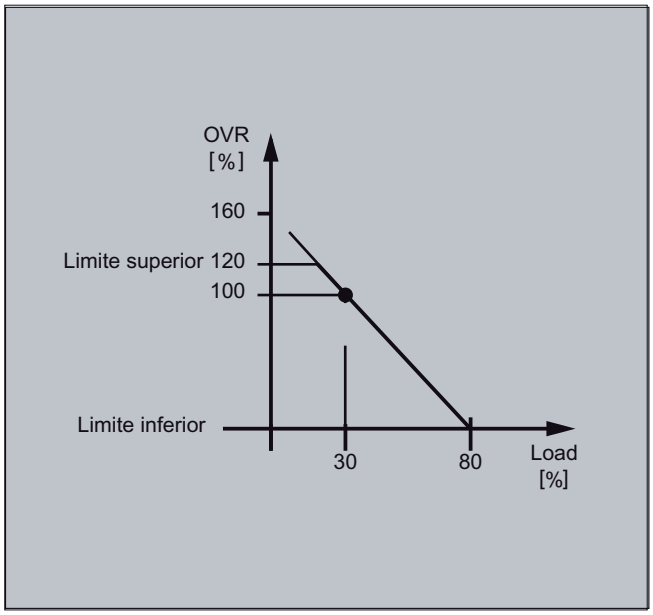
Exemplo de controle AC (multiplicativo)

Controlar o avanço programado de forma multiplicativa

O avanço programado deve ser controlado de forma multiplicativa, onde em função do avanço determinados limites de carga do acionamento não devem ser ultrapassados:

- O avanço deve parar com uma carga de acionamento de 80%: Override = 0.
- O avanço programado pode ser executado com uma carga de acionamento de 30%: Override = 100%.

A velocidade de avanço somente pode ser ultrapassada em até 20%:
Override máx. = 120%.



1. Definição de polinômio

Definição dos coeficientes

$y = f(x) = a_0 + a_1x + a_2x^2 + a_3x^3$

$a_1 = -100\%/(80-30)\% = -2$

$a_0 = 100 + (2*30) = 160$

$a_2 = a_3 = 0$ (nenhum elemento ao quadrado e ao cubo)

Limite superior = 120

Limite inferior = 0

Disso resulta:

```
FCTDEF(2,0,120,160,-2,0,0)
```

2. Ativar o controle AC

```
ID=1 DO SYNFCT(2,$AC_OVR,$AA_LOAD[x])
```

;Através do `$AA_LOAD[x]` ler a atual carga de eixo (% máx. da corrente de acionamento),
;calcular o Override de avanço com o polinômio definido acima.

10.4.8 Controle de distância com correção limitada ($AA_OFF_MODE)

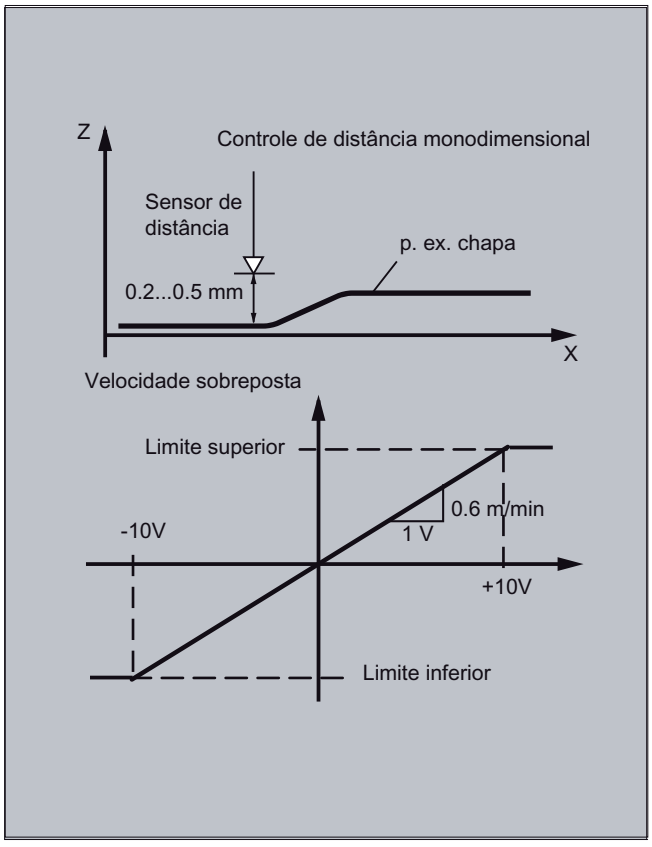
Indicação

Esta função não está disponível para o SINUMERIK 828D!

Função

O cálculo integrado dos valores de distância é realizado com controle de área limite:

$AA_OFF_MODE = 1



ATENÇÃO
A amplificação do circuito de controle sobreposto depende do ajuste do ciclo de interpolação. Solução: Leitura e inclusão do MD para ciclo de interpolação.

Indicação

Limitação da velocidade do interpolador sobreposto pelo MD32020 JOG_VELO com ciclo de interpolação de 12 ms.

Fórmula para velocidade:

$$\frac{0.120\text{mm}}{0.6\text{ms}} / \text{mV} = 0.6 \frac{\text{m}}{\text{min}} / \text{V}$$

Exemplo

Subrotina "AON": Controle de distância ON

```
Código de programa                                  Comentário
PROC AON
$AA_OFF_LIMIT[Z]=1                                  ; Definição de valor limite.
FCTDEF(1, -10, +10, 0, 0.6, 0.12)                   ; Definição de polinômio
ID=1 DO SYNFCT(1,$AA_OFF[Z],$A_INA[3])              ; Controle de distância ativo.
ID=2 WHENEVER $AA_OFF_LIMIT[Z]<>0                   ; Bloqueia o eixo X se a área
 DO $AA_OVR[X] = 0                                    limite for ultrapassada.
RET
ENDPROC
```

Subrotina "AOFF": Controle de distância OFF

```
Código de programa          Comentário
PROC AOFF
CANCEL(1)                   ; Cancelamento da ação síncrona do controle de distância
CANCEL(2)                   ; Cancelamento da verificação da área limite
RET
ENDPROC
```

Programa principal "MAIN"

```
Código de programa          Comentário
AON                         ; Controle de distância ON
...
G1 X100 F1000
AOFF                        ; Controle de distância OFF
M30
```

Outras informações

Deslocamento de posição no sistema de coordenadas básico

Com a variável de sistema SAA_OFF[eixo] é possível realizar um movimento sobreposto para cada eixo no canal. Ela atua como deslocamento de posição no sistema de coordenadas básico.

O deslocamento de posição programado é imediatamente sobreposto no eixo correspondente, independentemente se pelo programa o eixo é movimentado ou não.

Limitar a saída da variável de processamento principal:

É possível limitar o valor a ser corrigido de forma absoluta (saída de variável de processamento principal) no valor armazenado no dado de ajuste SD43350: $SA_AA_OFF_LIMIT.

O tipo de sobreposição da distância é definido através do dado de máquina MD36750: $MA_AA_OFF_MODE:

Valor	Significado
0	Avaliação proporcional
1	Avaliação integrante

Com a variável de sistema $AA_OFF_LIMIT[eixo] se pode consultar, em função do sentido, se o valor de correção encontra-se na área limite. Esta variável de sistema pode ser consultada a partir de ações sincronizadas e, ao ser alcançado um valor limite se pode, por exemplo, parar o eixo ou emitir um alarme.

0: Valor de correção não está na área limite
1 Foi alcançado o limite do valor de correção no sentido positivo
-1: Foi alcançado o limite do valor de correção no sentido negativo

10.4.9 Correção Online da ferramenta (FTOC)

Função

O FTOC permite um movimento sobreposto de um eixo geométrico conforme um polinômio programado com FCTDEF em função de um valor de referência que pode, por exemplo, ser o valor real de um eixo.

O coeficiente a_0 da definição de função FCTDEF(...) é avaliado com FTOC. Os limites superior e inferior dependem do a_0.

Com o FTOC as correções de ferramenta Online modais ou controles de distância podem ser programados como ações síncronas.

A função encontra aplicação na usinagem da peça de trabalho e dressagem do rebolo no mesmo canal ou em canais diferentes (canal de usinagem e canal de dressagem).

As condições gerais e definições para dressagem de rebolos são aplicadas para o FTOC de forma similar à correção de ferramenta Online com PUTFTOCF (veja "Correção de ferramenta Online (PUTFTOCF, FCTDEF, PUTFTOC, FTOCON, FTOCOF) (Página 415)").

Sintaxe

```
FCTDEF(<função>,<LLimit>,<ULimit>,<a0>,<a1>,<a2>,<a3>)
FTOC(<função>,<valor de referência>,<parâmetro de ferramenta>,<canal>,<fuso>)
...
```

Significado

FCTDEF: Com o FCTDEF é definida a função de polinômio para FTOC.

Parâmetro:

<função>:	Número da função de polinômio
	Tipo: INT
	Faixa de valores: 1 ... 3
<LLimit>:	Valor limite inferior
	Tipo: REAL
<ULimit>:	Valor limite superior
	Tipo: REAL
<a0> ... <a3>:	Coeficientes da função de polinômio
	Tipo: REAL

DO FTOC: Execução da função "Gravação contínua modal da correção de ferramenta Online"

Parâmetro:

<função>: Número da função de polinômio
Tipo: INT
Faixa de valores: 1 ... 3
Nota:
Precisa coincidir com a especificação no FCTDEF.

<valor de referência>: Variável de processamento principal, para a qual deve ser calculado um valor de função através da função de polinômio definida com FCTDEF.
Tipo: VAR REAL

<parâmetro de ferramenta>: Número do parâmetro de desgaste (comprimento 1, 2 ou 3), onde o valor de correção deve ser adicionado.
Tipo: INT

<canal>: Número do canal, onde a correção de ferramenta Online deve ser ativada.
Tipo: INT
Nota:
Uma especificação somente é necessária se a correção não deve ser ativada no canal ativo.

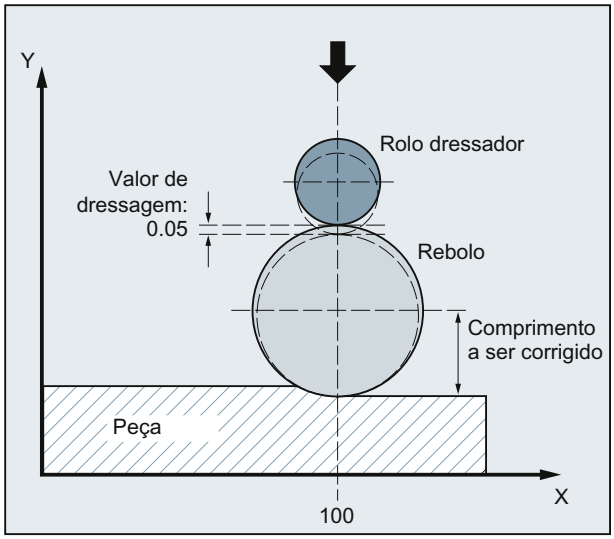
<fuso>: Número do fuso, onde a correção de ferramenta Online deve ser ativada.
Tipo: INT
Nota:
Uma especificação somente é necessária se, ao invés do rebolo ativo, deve ser corrigido um rebolo não ativo no momento da ferramenta empregada.

Indicação

No canal de destino deve estar ativado o FTOCON.

Exemplo

O comprimento do rebolo ativo e em uso deve ser corrigido.



Código de programa	**Comentário**
`FCTDEF(1,-1000,1000,-$AA_IW[V],1)`	`; Definição de função.`
`ID=1 DO FTOC(1,$AA_IW[V],3,1)`	`; Seleção da correção de ferramenta Online: Valor real do eixo V é o valor de entrada para o polinômio 1. O resultado é adicionado ao canal 1 como valor de correção do comprimento 3 do rebolo ativo.`
`WAITM(1,1,2)`	`; Sincronização com canal de usinagem.`
`G1 V-0.05 F0.01 G91`	`; Movimento de penetração para dressagem.`
`G1 V-0.05 F0.02`	
`...`	
`CANCEL(1)`	`; Cancelamento da correção Online.`
`...`	

10.4.10 Correção Online de comprimento de ferramenta ($AA_TOFF)

Função

Através da variável de sistema $AA_TOFF[ ] os comprimentos efetivos de ferramenta podem ser sobrepostos em tempo real de acordo com os três sentidos da ferramenta.

Como índice são utilizados os três identificadores de eixo geométrico. Com isso se define o número de sentidos de correção ativos através dos eixos geométricos durante o mesmo tempo.

Todas correções podem estar ativas simultaneamente.

Sintaxe

```
N... TRAORI
N... TOFFON(X,<valor Offset>)
N... WHEN TRUE DO $AA_TOFF[X]
N... TOFFON(Y,<valor Offset>)
N... WHEN TRUE DO $AA_TOFF[Y]
N... TOFFON(Z,<valor Offset>)
N... WHEN TRUE DO $AA_TOFF[Z]
```

Significado

`TOFFON`:	**Ativação** da correção de comprimento de ferramenta Online	
	`X, Y, Z` :	Sentido de ferramenta em que a correção de comprimento de ferramenta Online deve ter efeito.
	`<valor Offset>`:	Na ativação se pode especificar um valor de Offset para o respectivo sentido de correção que será imediatamente executado.
`TOFFOF`:	**Resetamento** da correção de comprimento de ferramenta Online Os valores de correção no sentido de correção especificado são resetados e se ativa uma parada de pré-processamento.	
`$AA_TOFF[X]=<valor>`:	Sobreposição no sentido X	
`$AA_TOFF[Y]=<valor>`:	Sobreposição no sentido Y	
`$AA_TOFF[Z]=<valor>`:	Sobreposição no sentido Z	

Exemplos

Exemplo 1: Seleção da correção de comprimento da ferramenta

Código de programa	Comentário
N10 TRAORI(1)	; Transformação ativada.
N20 TOFFON(Z)	; Ativação da correção Online do comprimento da ferramenta para o sentido Z da ferramenta.
N30 WHEN TRUE DO $AA_TOFF[Z]=10 G4 F5	; Para o sentido Z da ferramenta é interpolada uma correção do comprimento da ferramenta de 10
N40 TOFFON(X)	; Ativação da correção Online do comprimento da ferramenta para o sentido X da ferramenta.
N50 ID=1 DO $AA_TOFF[X]=$AA_IW[X2] G4 F5	; Para o sentido X da ferramenta é executada uma correção em função da posição do eixo X2.
...	
	; Atribuir atual correção no sentido X. Para o sentido X da ferramenta a correção do comprimento da ferramenta é novamente retornada como 0.
N100 XOFFSET=$AA_TOFF_VAL[X] N120 TOFFON(X, -XOFFSET) G4 F5	

Exemplo 2: Desativação da correção do comprimento da ferramenta

Código de programa	Comentário
N10 TRAORI(1)	; Transformação ativada.
N20 TOFFON(X)	; Ativação da correção Online do comprimento da ferramenta para o sentido X da ferramenta.
N30 WHEN TRUE DO $AA_TOFF[X] = 10 G4 F5	; Para o sentido X da ferramenta é interpolada uma correção do comprimento da ferramenta de 10.
...	
N80 TOFFOF(X)	; O Offset de posição do sentido X da ferramenta é deletado: ...$AA_TOFF[X]=0 Não é movimentado nenhum eixo. Para a atual posição em WCS é adicionado o deslocamento de posição conforme a atual orientação.

10.4.11 Movimentos de posicionamento

Função

Os eixos podem perfeitamente ser posicionados de forma assíncrona ao programa de peça a partir de ações sincronizadas. A programação dos eixos de posicionamento das ações sincronizadas é recomendada para processos cíclicos ou processos extremamente dependentes dos eventos. Os eixos programados a partir de ações sincronizadas são chamados de **eixos de comando**.

Programação

Literatura:
/PG/ Manual de programação Fundamentos; Capítulo "Dados de curso"
/FBSY/ Descrição do funcionamento Ações sincronizadas; "Iniciar eixos de comando"

Parâmetros

O sistema de medição para tarefas de posicionamento em ações sincronizadas se define com os códigos G `G70`/`G71`/`G700`/`G710` .

Com a programação de funções G na ação sincronizada se define a avaliação INCH/METRIC para a ação sincronizada independentemente do contexto do programa de peça.

10.4.12 Posicionar eixo (POS)

Função

Em comparação à programação a partir do programa de peça, o movimento do eixo de posicionamento não tem nenhum efeito sobre a execução do programa de peça.

Sintaxe

```
POS[eixo] = valor
```

Significado

`DO POS`	Partida/posicionamento de eixo de comando
`Eixo`	Nome do eixo que deve ser movimentado
`Valor`	Especificação do valor a ser percorrido (em função do modo de deslocamento)

Exemplos

Exemplo 1:

```
Código de programa                                    Comentário
ID=1 EVERY $AA_IM[B]>75 DO POS[U]=100                 ; O eixo U é movimentado de forma incremental a partir
                                                        do ponto zero do comando com 100 (pol./mm) ou até a
                                                        posição 100 (pol./mm), independentemente do modo de
                                                        deslocamento.
                                                      ; Eixo U movido no curso calculado a partir das
                                                        variáveis de processamento principal.
ID=1 EVERY $AA_IM[B]>75 DO POS[U]=$AA_MW[V]-$AA_IM[W]+13.5
```

Exemplo 2:

O ambiente de programa influencia o curso de posicionamento do eixo de posicionamento (nenhuma função G na parte da ação sincronizada):

```
Código de programa                                    Comentário
N100 R1=0
N110 G0 X0 Z0
N120 WAITP(X)
N130 ID=1 WHENEVER $R==1 DO POS[X]=10
N140 R1=1
N150 G71 Z10 F10                                      ; Z=10mm X=10mm
N160 G70 Z10 F10                                      ; Z=254mm X=254mm
N170 G71 Z10 F10                                      ; Z=10mm X=10mm
N180 M30
```

O G71 na parte da ação sincronizada define o curso de posicionamento do eixo de posicionamento de forma única (métrica), independentemente do ambiente de programa:

```
Código de programa                                    Comentário
N100 R1=0
N110 G0 X0 Z0
N120 WAITP(X)
N130 ID=1 WHENEVER $R==1 DO G71 POS[X]=10
N140 R1=1
N150 G71 Z10 F10                                      ; Z=10mm X=10mm
N160 G70 Z10 F10                                      ; Z=254mm X=10mm (X sempre posiciona em 10mm)
N170 G71 Z10 F10                                      ; Z=10mm X=10mm
N180 M30
```

Se o movimento do eixo não deve ser iniciado com o início do bloco, o override do eixo pode ser retido em 0 a partir de uma ação sincronizada até que o momento de partida desejado.

```
Código de programa                                    Comentário
WHENEVER $A_IN[1]==0 DO $AA_OVR[W]=0 G01 X10 Y25 F750 POS[W]=1500 FA=1000
                                                      ; O eixo de posicionamento permanece parado enquanto a
                                                        entrada digital 1=0.
```

10.4.13 Posição na área de referência especificada (POSRANGE)

Função

Com a função POSRANGE( ) se pode determinar se a atual posição nominal interpolada de um eixo encontra-se em uma janela de uma posição de referência especificada. As indicações de posição podem se referir aos sistemas de coordenadas especificáveis.

Na consulta de uma posição real de eixo de um eixo Modulo se considera a correção Modulo.

Indicação

A função somente pode ser chamada a partida ação sincronizada. Na chamada a partir do programa de peça ocorre o alarme 14091 %1 Bloco %2 Função não admissível, índice: %3 chamado com o índice 5.

Sintaxe

```
BOOL POSRANGE(eixo, Refpos, Winlimit,[Coord])
```

Significado

`BOOL POSRANGE`	A atual posição do eixo de comando está na janela da posição de referência especificada.
`AXIS <eixo>`	Identificador de eixo de máquina, canal ou geométrico
`REAL Refpos`	Posição de referência no sistema de coordenadas Coord
`REAL Winlimit`	Valor que resulta no limite para janela de posição
`INT Coord`	Opcionalmente o MCS está ativo. São possíveis: 0 para MCS (sistema de coordenadas da máquina) 1 para BCS (sistema de coordenadas básico) 2 para ENS (sistema de ponto zero ajustável) 3 para WCS (sistema de coordenadas da peça)

Valor de função

Atual posição nominal em função da indicação de posição no sistema de coordenadas especificado

`Valor de função: TRUE`	se Refpos(Coord) - abs(Winlimit) ≤ Actpos(Coord) ≤ Refpos(Coord) + abs(Winlimit)
`Valor de função: FALSE`	senão

10.4.14 Iniciar/parar eixo (MOV)

Função

Com o MOV[eixo]=valor pode-se iniciar um eixo de comando sem indicação de uma posição final. O respectivo eixo é deslocado no sentido programado até ser especificado outro movimento por um novo comando de movimento ou de posicionamento, ou o eixo ser parado com um comando de parada.

Sintaxe

```
MOV[eixo] = valor
```

Significado

`DO MOV`	Iniciar movimento do eixo de comando
`Eixo`	Nome do eixo que deve ser iniciado
`Valor`	Comando para movimento de deslocamento e de parada O sinal determina o sentido do movimento O tipo de dado do valor é INTEGER.
`Valor >0 (normalmente +1)`	sentido positivo
`Valor <0 (normalmente -1)`	sentido negativo
`Valor ==0`	Parar movimento do eixo

Indicação

Se um eixo de indexação for parado com `MOV[eixo] = 0` , o eixo será parado na próxima posição de indexação.

Exemplo

Código de programa	**Comentário**
`... DO MOV[U]=0`	`;  O eixo U é parado`

10.4.15 Troca de eixos (RELEASE, GET)

Função

Para uma troca de ferramentas podem ser solicitados os respectivos eixos de comando como ação de uma ação sincronizada com GET(eixo). O tipo de eixo atribuído à este canal e, com isso, o direito de interpolação associado neste momento, pode ser consultado através da variável de sistema $AA_AXCHANGE_TYP. Em função do próprio estado e do canal que o atual direito de interpolação deste eixo possui, é possível realizar diferentes processos.

Se a troca de ferramentas estiver concluída, então este eixo de comando pode ser liberado como ação de uma ação sincronizada com RELEASE(eixo) para o canal.

Fabricante da máquina

O eixo afetado deve estar associado ao canal através de dados de máquina. Observe as instruções do fabricante da máquina.

Sintaxe

`GET(eixo[,eixo{,...}])` Solicitar eixo

`RELAESE(eixo[,eixo{,...}])` Liberar eixo

Significado

`DO RELEASE`	Liberar eixo como eixo neutro
`DO GET`	Buscar eixo para troca de eixos
`Eixo`	Nome do eixo que deve ser iniciado

Exemplo de execução de programa para uma troca de eixos de dois canais

O eixo Z é conhecido no 1º canal e no 2º canal.

Execução de programa no 1º canal:

```
Código de programa                              Comentário
WHEN TRUE DO RELEASE(Z)                         ; O eixo Z passa a ser eixo neutro.
WHENEVER($AA_TYP[Z]==1) DO RDISABLE             ; Bloqueio de leitura enquanto o
                                                  eixo Z for eixo do programa
N110 G4 F0.1
WHEN TRUE DO GET(Z)                             ; O eixo Z passa a ser novamente
                                                  eixo de programa NC
WHENEVER($AA_TYP[Z]<>1) DO RDISABLE             ; Bloqueio de leitura até o eixo Z
                                                  ser eixo do programa
N120 G4 F0.1
WHEN TRUE DO RELEASE(Z)                         ; O eixo Z passa a ser eixo neutro.
WHENEVER($AA_TYP[Z]==1) DO RDISABLE             ; Bloqueio de leitura enquanto o
                                                  eixo Z for eixo do programa
N130 G4 F0.1                                    ;
N140 START(2)                                   ; iniciar o 2° canal
```

Execução de programa no 2º canal:

Código de programa	Comentário
WHEN TRUE DO GET(Z)	; ;Buscar eixo Z no 2° canal
WHENEVER($AA_TYP[Z]==0) DO RDISABLE	; ;Bloqueio de leitura enquanto o eixo Z estiver em ;outro canal
N210 G4 F0.1	
WHEN TRUE DO GET(Z)	; ;Eixo Z passa a ser eixo de programa NC
WHENEVER($AA_TYP[Z]<>1) DO RDISABLE	; ;Bloqueio de leitura até o eixo Z ser eixo do programa
N220 G4 F0.1	
WHEN TRUE DO RELEASE(Z)	; ;Eixo Z no 2° canal é eixo neutro
WHENEVER($AA_TYP[Z]==1) DO RDISABLE	; ;Bloqueio de leitura enquanto o eixo Z for eixo do programa
N230 G4 F0.1	
N250 WAITM(10, 1, 2)	; sincronizar com canal 1

Continuação da execução de programa no 1º canal:

Código de programa	Comentário
N150 WAIM(10, 1, 2)	; sincronizar com canal 2
WHEN TRUE DO GET(Z)	; Buscar eixo Z neste canal
WHENEVER($AA_TYP[Z]==0) DO RDISABLE	; Bloqueio de leitura enquanto o eixo Z estiver em outro canal
N160 G4 F0.1	
N199 WAITE(2)	
N999 M30	; Espera pelo fim do programa no canal 2

Exemplo de troca de eixos no ciclo de tecnologia

O eixo U ($MA_AUTO_GET_TYPE=2) é conhecido no 1º canal e no 2º canal e atualmente o canal 1 é que detém o direito de interpolação. No canal 2 é iniciado o seguinte ciclo de tecnologia:

Código de programa	Comentário
GET(U)	; Buscar eixo U no canal
POS[U]=100	; O eixo U deve ser deslocado até a posição 100

A linha do movimento de eixo de comando POS[U] somente será executada se o eixo U for buscado no canal 2.

Seqüência de operação

O eixo solicitado no momento de ativação da ação `GET(eixo)` pode ser lido quanto ao tipo de eixo para uma troca de eixos através da variável de sistema ($AA_AXCHANGE_TYP[<eixo>]:

- 0: Eixo associado ao programa NC
- 1: Eixo do PLC atribuído ou ativo como eixo de comando ou eixo de oscilação
- 2: um outro canal possui o direito de interpolação
- 3: O eixo é eixo neutro
- 4: eixo neutro controlado pelo PLC
- 5: um outro canal possui direito de interpolação, o eixo é solicitado para o programa NC
- 6: um outro canal possui direito de interpolação, o eixo é solicitado como eixo neutro
- 7: Eixo do PLC ativo como eixo de comando ou eixo oscilante, o eixo é solicitado para o programa NC
- 8: Eixo do PLC ativo como eixo de comando ou eixo oscilante, o eixo é solicitado como eixo neutro

Condições gerais

O eixo afetado deve estar associado ao canal através de dados de máquina.

Um eixo controlado exclusivamente pelo PLC não pode ser atribuído ao programa NC.

Literatura:

/FB2/ Manual de funções ampliadas; Eixos de posicionamento (P2)

Solicitar eixo de outro canal com a ação GET

Se no momento da ativação da ação `GET` um **outro canal tiver o direito de gravação** (direito de interpolação) para o eixo ($AA_AXCHANGE_TYP[<eixo>] == 2), então o eixo é solicitado através da troca deste canal ($AA_AXCHANGE_TYP[<eixo>]==6) e, assim que possível, associado ao canal solicitado.

Ele assume então o estado de eixo neutro ($AA_AXCHANGE_TYP[<eixo>]==3).

Não se realiza nenhuma reorganização no canal solicitado.

Atribuição **como eixo de programa NC com reorganização:**

Se o eixo foi solicitado no momento da ativação da ação `GET` como eixo neutro ($AA_AXCHANGE_TYP[<eixo>]==6), então o eixo é solicitado para o programa NC ($AA_AXCHANGE_TYP[<eixo>]==5) e, assim que possível, atribuído ao programa NC do canal ($AA_AXCHANGE_TYP[<eixo>]==0).

Eixo associado ao canal solicitado

Atribuição **como eixo de programa NC com reorganização:**

Se o eixo solicitado no momento da ativação já está associado ao canal solicitado, e em estado de eixo neutro – não controlado pelo PLC – ($AA_AXCHANGE_TYP[<eixo>]==3), então ele será atribuído ao programa NC ($AA_AXCHANGE_TYP[<eixo>]==0).

O eixo em estado de eixo neutro é controlado pelo PLC

Se o eixo está em estado de eixo neutro controlado pelo PLC ($AA_AXCHANGE_TYP[<eixo>]==4), então o eixo é solicitado como eixo neutro ($AA_AXCHANGE_TYP[<eixo>] == 8), neste caso, em função do Bit 0 no dado de máquina MD 10722: AXCHANGE_MASK o eixo é bloqueado para uma troca automática de eixos entre canais (Bit 0 == 0). Isto corresponde à ($AA_AXCHANGE_STAT[<eixo>] == 1).

O eixo está ativo como eixo neutro de comando ou de oscilação ou atribuído ao PLC

Se o eixo está ativo como eixo de comando ou oscilante ou está associado ao PLC para deslocamento, eixo PLC == eixo de posicionamento concorrente, ($AA_AXCHANGE_TYP[<eixo>]==1), então o eixo é solicitado como eixo neutro ($AA_AXCHANGE_TYP[<eixo>] == 8), neste caso, em função do Bit 0 no dado de máquina MD 10722: AXCHANGE_MASK o eixo é bloqueado para uma troca automática de eixos entre os canais (Bit 0 == 0). Isto corresponde à ($AA_AXCHANGE_STAT[<eixo>] == 1).

A nova ação GET solicita o eixo para o programa NC ($AA_AXCHANGE_TYP[<eixo>] será == 7).

O eixo já está associado ao programa NC

Se o eixo já se encontra atribuído ao programa NC do canal ($AA_AXCHANGE_TYP[<eixo>]==0) ou se esta atribuição foi solicitada, p. ex. acionada a troca de eixos do programa NC ($AA_AXCHANGE_TYP[<eixo>]==5 ou $AA_AXCHANGE_TYP[<eixo>] == 7), então não resulta em nenhuma alteração de estado.

10.4.16 Avanço por eixo (FA)

Função

O avanço por eixo para eixos de comando está ativo de forma modal.

Sintaxe

```
FA[<eixo>]=<valor>
```

Exemplo

Código de programa	Comentário
`ID=1 EVERY $AA_IM[B]>75 DO POS[U]=100 FA[U]=990`	`; Definir valor fixo de avanço.`
	`; Formar valor de avanço a partir de variáveis de processamento principal:`
`ID=1 EVERY $AA_IM[B]>75 DO POS[U]=100 FA[U]=$AA_VACTM[W]+100`	

10.4.17 Chave fim de curso de SW

Função

A limitação a área de trabalho programada com G25/G26 é considerada em função do dado de ajuste $SA_WORKAREA_PLUS_ENABLE para os eixos de comando.

A ativação e desativação da limitação da área de trabalho através das funções G WALIMON/WALIMOF no programa de peça não tem nenhum efeito sobre os eixos de comando.

10.4.18 Coordenação de eixo

Função

Tipicamente um eixo é movimentado ou a partir de um programa de peça ou como eixo de posicionamento a partir de uma ação sincronizada.

Entretanto, se este eixo deve ser movimentado alternadamente a partir do programa de peça como eixo de percurso ou eixo de posicionamento e a partir de ações sincronizadas, então é realizada uma transferência coordenadas entre ambos movimentos de eixo.

Se em seguida um eixo de comando é movimentado a partir de um programa de peça, então isto requer uma reorganização do pré-processamento. Por sua vez, isto requer uma interrupção do processamento do programa de peça, semelhante à uma parada de pré-processamento.

Exemplo de deslocamento do eixo X a partir do programa de peça ou a partir de ações sincronizadas

Código de programa	Comentário
N10 G01 **X100** Y200 F1000	; Eixo X programado no programa de peça
…	
N20 ID=1 WHEN $A_IN[1]==1 DO **POS[X]=150** FA[X]=200	; Iniciar o posicionamento a partir da ação sincronizada, ;se houver entrada digital
…	
CANCEL(1)	; Desselecionar ação sincronizada
…	
N100 G01 **X240** Y200 F1000	; X é eixo de percurso; antes do movimento ocorre um tempo de espera em função da transferência de eixo, se a entrada digital era 1 e X posicionado a partir da ação sincronizada.

Exemplo de alteração de comando de movimento para este mesmo eixo

Código de programa	Comentário
ID=1 EVERY $A_IN[1]>=1 DO POS[V]=100 FA[V]=560	; Iniciar posicionamento a partir da ação sincronizada, se a entrada digital >= 1
ID=2 EVERY $A_IN[2]>=1 DO POS[V]=$AA_IM[V] FA[V]=790	; O eixo acompanha, a 2ª entrada é definida, isto é, a posição final e o avanço do eixo V são continuamente acompanhados durante o movimento se duas ações sincronizadas estiverem ativas simultaneamente.

10.4.19 Definir valor real (PRESETON)

Função

Para a execução do PRESETON (eixo, valor) não é alterada a atual posição de eixo, à será atribuído um novo valor.

O PRESETON a partir de ações sincronizadas é possível para:

- Eixos rotativos Modulo que foram iniciados a partir do programa de peça
- Todos eixos de comando que foram iniciados a partir da ação sincronizada

Sintaxe

```
DO PRESETON(eixo, valor)
```

Significado

`DO PRESETON`	Definição de valor real em ações sincronizadas
`Eixo`	Eixo cujo ponto zero do comando deve ser alterado
`Valor`	Valor para o qual o ponto zero do comando deve ser alterado

Restrições para eixos

O `PRESETON` não é possível para eixos que estão envolvidos na transformação.

Um e o mesmo eixo somente pode ser movimentado a partir de um programa de peça ou de uma ação sincronizada se ocorrer em momentos diferentes, por isso que na programação de um eixo no programa de peça podem aparecer tempos de espera, caso este eixo tenha sido programado anteriormente em uma ação sincronizada.

Se o mesmo eixo for utilizado alternadamente, então ocorre uma transferência coordenada entre ambos movimentos do eixo. Para isso se deve interromper o processamento do programa de peça.

Exemplo

Deslocamento do ponto zero de comando de um eixo

```
Código de programa                                  Comentário
WHEN $AA_IM[a] >= 89.5 DO PRESETON(a4,10.5)     ;  Deslocamento do ponto zero do
                                                   comando do eixo a em 10.5
                                                   unidades de comprimento (polegada
                                                   ou mm)
                                                   no sentido positivo do eixo
```

10.4.20 Movimentos de fuso

Função

Os fusos podem perfeitamente ser posicionados de forma assíncrona ao programa de peça a partir de ações sincronizadas. Este tipo de programação é recomendada para processos cíclicos ou processos extremamente dependentes dos eventos.

Se forem especificados comandos concorrentes para um fuso através de ações sincronizadas ativas simultaneamente, se aplica o comando de fuso mais recente.

Exemplo da ação de iniciar/parar/posicionar o fuso

Código de programa	Comentário
`ID=1 EVERY $A_IN[1]==1 DO M3 S1000`	`; Ajustar o sentido de giro e rotação`
`ID=2 EVERY $A_IN[2]==1 DO SPOS=270`	`; Posicionar fuso`

Exemplo de ajuste do sentido de giro, rotação / posicionamento do fuso

Código de programa	Comentário
`ID=1 EVERY $A_IN[1]==1 DO M3 S300`	`; Ajustar o sentido de giro e rotação`
`ID=2 EVERY $A_IN[2]==1 DO M4 S500`	`; Especificar novo sentido de giro e nova rotação`
`ID=3 EVERY $A_IN[3]==1 DO S1000`	`; Especificar nova rotação`
`ID=4 EVERY ($A_IN[4]==1) AND ($A_IN[1]==0) DO SPOS=0`	`; Posicionar fuso`

10.4.21 Movimento acoplado (TRAILON, TRAILOF)

Função

O eixo mestre pode estar em movimento durante a ativação do acoplamento a partir da ação sincronizada. Neste caso o eixo escravo é acelerado até a velocidade nominal. A posição do eixo mestre no momento da ação sincronizada das velocidades é a posição de partida do movimento acoplado. A funcionalidade do movimento acoplado está descrita no capítulo "Comportamento de deslocamento do percurso".

Sintaxe

Ativar movimento acoplado

```
DO TRAILON(eixo escravo, eixo mestre, fator de acoplamento)
```

Desativar movimento acoplado

```
DO TRAILOF(eixo escravo, eixo mestre, eixo mestre 2)
```

Significado

Ativar movimento acoplado assíncrono:

`... DO TRAILON(FA, LA, Kf)`

com:
FA: Eixo escravo
LA: Eixo mestre
Kf: Fator de acoplamento

Desativar movimento acoplado assíncrono:

`... DO TRAILOF(FA, LA, LA2)`

com:
FA: Eixo escravo
LA: Eixo mestre, opcional
LA2: Eixo mestre 2, opcional

`... DO TRAILOF(FA)`

Todos acoplamentos para eixo escravo são desativados.

Exemplo

Código de programa		Comentário
`$A_IN[1]==0 DO TRAILON(Y,V,1)`	;	Ativação do 1° grupo de movimento acoplado, se a entrada digital for 1
`$A_IN[2]==0 DO TRAILON(Z,W,-1)`	;	Ativação do 2° grupo de movimento acoplado
`G0 Z10`	;	Penetração do eixo Z e eixo W no sentido oposto do ;eixo
`G0 Y20`	;	Penetração do eixo Y e eixo V no mesmo sentido do eixo
`...`		
`G1 Y22 V25`	;	Sobreposição de um movimento dependente e um independente ;do eixo acoplado "V"
`...`		
`TRAILOF(Y,V)`	;	Desativação do 1° grupo de movimento acoplado
`TRAILOF(Z,W)`	;	Desativação do 2° grupo de movimento acoplado

Exemplo para evitar de conflito com TRAILOF

Para liberar novamente um eixo acoplado para o acesso como eixo de canal, deve-se chamar primeiramente a função `TRAILOF`. Deve-se assegurar que o `TRAILOF` foi executado antes do canal solicitar o respectivo eixo. Este não é o caso no seguinte exemplo

```
…
N50 WHEN TRUE DO TRAILOF(Y,X)
N60 Y100
…
```

Neste caso o eixo não é liberado a tempo, pois a ação sincronizada ativa por blocos com `TRAILOF` é ativada sincronizadamente com `N60`, veja o capítulo "Sincronização de movimentos, estrutura", fundamentos gerais.
Para evitar situações de conflito se deve

proceder da seguinte forma

```
…
N50 WHEN TRUE DO TRAILOF(Y,X)
N55 WAITP(Y)
N60 Y100
```

10.4.22 Acoplamento de valor mestre (LEADON, LEADOF)

Indicação

Esta função não está disponível para o SINUMERIK 828D!

Função

O acoplamento axial de valor mestre é programável sem restrições em ações sincronizadas. A alteração de uma tabela de curvas no acoplamento existente sem uma nova sincronização anterior somente é possível como opcional em ações sincronizadas.

Sintaxe

Ativação do acoplamento de valor mestre

```
DO LEADON(eixo escravo, eixo mestre, tab. de curva n°, OVW)
```

Desativação do acoplamento de valor mestre

```
DO LEADOF(eixo escravo, eixo mestre, eixo mestre 2)
```

Significado

`Ativação do acoplamento de valor mestre axial:`	
`...DO LEADON(FA, LA, NR, OVW)`	com: FA: Eixo escravo LA: Eixo mestre NR: Número da tabela de curvas armazenada OVW: Permissão de sobregravação de um acoplamento existente com tabela de curvas alterada
`Desativação do acoplamento de valor mestre axial:`	
`...DO LEADOF(FA, LA)`	com: FA: Eixo escravo LA: Eixo mestre, opcional
`... DO LEADOF(FA)`	formato abreviado sem indicação do eixo mestre

Liberação do acesso por ações síncronas RELEASE

Para que um eixo a ser acoplado seja liberado ao acesso através de ação síncrona, deve-se chamar primeiramente a função RELEASE para o eixo escravo a ser acoplado.

Exemplo:

```
RELEASE(XKAN)
ID=1 every SR1==1 to LEADON(CACH,XKAN,1)
```

OVW=0 (valor padrão)

Para um acoplamento existente não se pode especificar nenhuma nova tabela de curvas sem uma nova sincronização. Uma alteração da tabela de curvas requer a desativação anterior do acoplamento existente e uma nova ativação com o novo número de tabela de curvas. Isto resulta numa nova sincronização do acoplamento.

Alteração da tabela de curvas com acoplamento existente com OVW=1

Com `OVW=1` se pode especificar uma nova tabela de curvas para um acoplamento existente. Não ocorre nenhuma nova sincronização. O eixo escravo tenta o mais rápido possível acompanhar os valores de posição especificados pela nova tabela de curvas.

Exemplo de separação flutuante

Um material extrudado que se movimenta continuamente pela área de trabalho através de um dispositivo de corte deve ser dividido em partes de comprimento igual.

Eixo X: Eixo em que o material extrudado se movimenta. WCS
Eixo X1: Eixo de máquina do material extrudado, MCS
Eixo Y: Eixo com o qual o dispositivo de corte "acompanha" o material extrudado

Considera-se que a penetração da ferramenta de separação e seu comando sejam controlados via PLC. Para definição do sincronismo entre material extrudado e ferramenta de separação podem ser avaliados os sinais da interface do PLC.

Ações
Ativar acoplamento, `LEADON`
Desativar acoplamento, `LEADOF`
Definir valor real, `PRESETON`

Código de programa	**Comentário**
N100 R3=1500	; Comprimento de uma peça a ser cortada
N200 R2=100000 R13=R2/300	
N300 R4=100000	
N400 R6=30	; Posição de partida do eixo Y
N500 R1=1	; Condição de partida para eixo de transporte
N600 LEADOF(Y,X)	; Deletação de um eventual acoplamento existente
N700 CTABDEF(Y,X,1,0)	; Definição de tabela
N800 X=30 Y=30	; Pares de valores
N900 X=R13 Y=R13	
N1000 X=2*R13 Y=30	
N1100 CTABEND	; Fim da definição de tabela
N1200 PRESETON(X1,0)	; PRESET para iniciar
N1300 Y=R6 G0	; Posição de partida do eixo Y, eixo é linear
N1400 ID=1 WHENEVER $AA_IW[X]>$R3 DO PESETON(X1,0)	; PRESET após comprimento R3, reinício após separação
N1500 RELEASE(Y)	
N1800 ID=6 EVERY $AA_IM[X]<10 DO LEADON(Y,X,1)	; Acopla Y ao X através da tabela 1, com X < 10
N1900 ID=10 EVERY $AA_IM[X]>$R3-30 DO EADOF(Y,X)	; > 30 desacopla antes do comprimento de separação percorrido
N2000 WAITP(X)	
N2100 ID=7 WHEN $R1==1 DO MOV[X]=1 FA[X]=$R4	; Definição de eixo do material extrudado em movimento contínuo
N2200 M30	

10.4.23 Medição (MEAWA, MEAC)

Função

Comparado com a utilização de blocos de movimento do programa de peça, a função de medição pode ser ativada e desativada a partir de ações sincronizadas quando necessário.

Para mais informações sobre medição, veja a função ampliada de medição de comandos de curso especiais

Sintaxe

Medição por eixo sem anulação do curso restante

```
MEAWA[eixo] = (modo, evento de ativação_1, ..._4)
```

Medição constante sem anulação de curso restante

```
MEAC[eixo] = (modo, memória de medição, evento de ativação_1, ..._4))
```

Significado

Código de programa	**Comentário**	
DO MEAWA	; Ativar medição axial	
DO MEAC	; Ativar medição contínua	
Eixo	; Nome do eixo que deve ser medido	
Modo	; Indicação da casa da **dezena** 0: sistema de medição ativo Número de sistemas de medição (em função do modo) 1: 1. Sistema de medição 2: 2. Sistema de medição 3: ambos sistemas de medição	Indicação da casa da **unidade** 0: Cancelar pedido de medição até 4 eventos de ativação ativados sucessivamente 1: simultaneamente 2: sucessivamente 3: como 2, mas sem monitoração do evento de ativação 1 na partida
evento de ativação_1 até _4	; : Flancos crescentes do apalpador de medição 1 -1: Flancos decrescentes do apalpador de medição 1 opcional 2: Flancos crescentes do apalpador de medição 2 opcional -2: Flancos decrescentes do apalpador de medição 2 opcional	
Memória de medição	; Número da memória de circulação FIFO	

10.4.24 Inicialização de variáveis de campo (SET, REP)

Função

Em ações sincronizadas as variáveis de campo podem ser inicializadas ou com descritas com determinados valores.

Indicação

Somente são possíveis variáveis que podem ser descritas em ações sincronizadas. Desse modo os dados de máquina não permitem ser inicializados. As variáveis de eixo não podem ser especificadas com o valor NO_AXIS.

Sintaxe

```
DO CAMPO[n,m]=SET(<valor1>,<valor2>,...)
DO CAMPO[n,m]=REP(<valor>)
```

Significado

`CAMPO[n,m]`	Índice de campo programado A inicialização inicia no índice de campo programado. Em campos de 2 dimensões se incrementa primeiro o 2º índice. O índice de eixo não é processado neste caso.
`SET(<valor1>,<valor2>,...)`	Inicialização com listas de valores O campo é descrito pelo índice de campo programado com os parâmetros do `SET`. São atribuídos tantos elementos de campo como valores programados. Se forem programados mais valores do que elementos de campo disponíveis, o sistema emitirá um alarme
`REP(<valor>)`	Inicialização com mesmos valores O campo é novamente descrito a partir do índice de campo programado até o fim do campo com os parâmetros (`<valor>`) do `REP`.

Exemplo

Código de programa	**Comentário**
`WHEN TRUE DO SYG_IS[0]=REP(0)`	`; Resultado:`
`WHEN TRUE DO SYG_IS[1]=SET(3,4,5)`	`; SYG_IS[0]=0`
	`SYG_IS[1]=3`
	`SYG_IS[2]=4`
	`SYG_IS[3]=5`
	`SYG_IS[4]=0`

10.4.25 Definir/deletar marcadores de espera (SETM, CLEARM)

Função

Em ações sincronizadas podem ser definidos ou deletados marcadores de espera, por exemplo, para coordenar canais entre si.

Sintaxe

```
DO SETM(<número do marcador>)
DO CLEARM(<número do marcador>)
```

Significado

`SETM`	Comando para definir o marcador de espera para o canal O comando `SETM` pode ser escrito no programa de peça e na parte de ação de uma ação sincronizada. Ele define o marcador (`<número do marcador>`) para o canal em que o comando é processado.
`CLEARM`	Comando para deletar o marcador de espera para o canal O comando `CLEARM` pode ser escrito no programa de peça e na parte de ação de uma ação sincronizada. Ele deleta o marcador (`<número do marcador>`) para o canal em que o comando é processado.
`<número do marcador>`	Marcador de espera

10.4.26 Reações à erros (SETAL)

Função

Com ações sincronizadas podem ser programadas reações à erros. Neste caso são consultadas variáveis de estado e executadas as ações correspondentes.

As possíveis reações à estados de erro são:

- Parada de eixo (Override=0)
- Definição de alarme

  Com SETAL podem ser definidos alarmes de ciclos a partir de ações sincronizadas.
- Definição de saída
- Todas ações possíveis em ações sincronizadas

Sintaxe

Definição de alarme de ciclo:

```
DO SETAL(<número de alarme>)
```

Significado

SETAL	Comando para definir um alarme de ciclo
<número de alarme>	Número do alarme Faixa de alarmes de ciclo para o usuário: 65000 até 69999

Exemplo

Código de programa	Comentário
`ID=67 WHENEVER ($AA_IM[X1]-$AA_IM[X2])<4.567 DO $AA_OVR[X2]=0`	`; Se a distância de segurança entre os eixos X1 e X2 for muito pequena, parar eixo X2.`
`ID=67 WHENEVER ($AA_IM[X1]-$AA_IM[X2])<4.567 DO SETAL(65000)`	`; Se a distância de segurança entre os eixos X1 e X2 for muito pequena, definir alarme 65000.`

10.4.27 Deslocar até o encosto fixo (FXS, FXST, FXSW, FOCON, FOCOF)

Função

Os comandos para a função de "deslocamento até o encosto fixo" são programados em ações sincronizadas/ciclos de tecnologia com os comandos de programa de peça `FXS`, `FXST` e `FXSW`.

A ativação pode ser realizada sem movimento, o torque é imediatamente limitado. Assim que o eixo for movimentado com o valor nominal, será realizada a monitoração no encosto.

Deslocamento com torque/força limitada (FOC)

A função permite alterar o torque/força a qualquer momento através de ações sincronizadas e pode ser ativada modalmente ou por blocos.

Sintaxe

```
FXS[<eixo>]
FXST[<eixo>]
FXSW[<eixo>]
FOCON[<eixo>]
FOCOF[<eixo>]
```

Significado

`FXS`	Seleção somente em sistemas com acionamentos digitais (VSA, HSA, HLA)
`FXST`	Alteração do momento de fixação FXST
`FXSW`	Alteração da janela de monitoração FXSW
`FOCON`	Ativação da limitação de torque/força ativa modalmente
`FOCOF`	Desativação da limitação de torque/força
<eixo>	Identificador de eixo São permitidos: • Identificador de eixo geométrico • Identificador de eixo de canal • Identificador de eixo de máquina

Indicação

Uma seleção somente pode ser realizada uma vez.

Exemplos

Exemplo 1: Deslocamento até o encosto fixo (FXS), acionado através de uma ação síncrona

```
Código de programa                                                    Comentário
Eixo Y:                                                               ; Ações sincronizadas estáticas
ativar:
N10 IDS=1 WHENEVER (($R1==1) AND $AA_FXS[y]==0)) D $R1=0 FXS[Y]=1 FXST[Y]=10 FA[Y]=200 POS[Y]=150
                                                                      ; Com a definição de $R1=1 é ativado o FXS
                                                                        para o eixo Y, o torque ativo é reduzido
                                                                        até 10% e iniciado um movimento no
                                                                        sentido do encosto.
N11 IDS=2 WHENEVER ($AA_FXS[Y]==4) DO FXST[Y]=30                      ; Assim que o encosto dor detectado
                                                                        ($AA_FXS[Y]==4), o torque será elevado
                                                                        até 30%.
N12 IDS=3 WHENEVER ($AA_FXS[Y]==1) DO FXST[Y]=$R0                     ; Após alcançar o encosto o torque é
                                                                        controlado em função do R0.
N13 IDS=4 WHENEVER (($R3==1) AND $AA_FXS[Y]==1)) DO FXS[Y]=0 FA[Y]=1000 POS[Y]=0
                                                                      ; Desseleção em função do R3 e recuo.
N20 FXS[Y]=0 G0 G90 X0 Y0                                             ; Execução normal do programa:
N30 RELEASE(Y)                                                        ; Liberação do eixo Y para o movimento na
                                                                        ação síncrona.
N40 G1 F1000 X100                                                     ; Movimento de outro eixo.
N50 ...
N60 GET(Y)                                                            ; Eixo Y novamente incluso no grupo de
                                                                        percurso
```

Exemplo 2: Ativação da limitação de torque/força (FOC)

```
Código de programa                  Comentário
N10 FOCON[X]                        ; Ativação modal da limitação.
N20 X100 Y200 FXST[X]=15            ; X desloca-se com torque reduzido (15%).
N30 FXST[X]=75 X20                  ; Alteração do torque para 75%, X desloca-se com
                                      este torque limitado.
N40 FOCOF[X]                        ; Desativação da limitação de torque.
```

Outras informações

Seleção múltipla

Se por causa de uma programação incorreta a função for novamente chamada após a ativação (`FXS[<eixo>]=1`), será emitido o seguinte alarme:

Alarme 20092 "Deslocamento até o encosto fixo ainda ativo"

Uma programação que na condição consulta o $AA_FXS[ ] ou um próprio marcador (aqui é o R1) evita uma ativação múltipla da função "Fragmento de programa de peça":

```
Código de programa
N10 R1=0
N20 IDS=1 WHENEVER ($R1==0 AND
$AA_IW[AX3] > 7) DO R1=1 FXST[AX1]=12
```

Ações sincronizadas relativa aos blocos

Com a programação de uma ação sincronizada relativa ao blocos, o deslocamento até o encosto fixo pode ser conectado durante um movimento de aproximação.

Exemplo:

Código de programa	Comentário
N10 G0 G90 X0 Y0	
N20 WHEN $AA_IW[X] > 17 DO FXS[X]=1	; Se X alcança uma posição maior que 17 mm, então o FXS é ativado.
N30 G1 F200 X100 Y110	

Ações sincronizadas estáticas e relativas ao bloco

No caso das ações sincronizadas estáticas e relativas ao blocos podem ser utilizados os mesmos comandos `FXS`, `FXST` e `FXSW` como no caso normal de execução do programa de peça. Os valores que são atribuídos podem ter sido obtidos através de um cálculo.

10.4.28 Definição do ângulo de tangente de percurso em ações sincronizadas

Função

A variável de sistema $AC_TANEB (**T**angent **AN**gel at **E**nd of **B**lock) que pode ser lida em ações sincronizadas determina o ângulo entre a tangente de percurso no ponto final do atual bloco e a tangente de percurso no ponto de partida do bloco subseqüente programado.

Parâmetros

O ângulo de tangente sempre é informado positivo na faixa de 0.0 até 180.0 graus. Se não houver nenhum bloco seguinte, então é retornado o ângulo de -180.0 graus.

A variável de sistema `$AC_TANEB` não deveria ser lida para blocos que foram gerados pelo sistema (blocos intermediários). A variável de sistema `$AC_BLOCKTYPE` serve para saber quando se trata de um bloco programado (bloco principal).

Exemplo

```
ID=2 EVERY $AC_BLOCKTYPE==0 DO $SR1 = $AC_TANEB
```

10.4.29 Determinação do atual Override

Função

O atual override

(Parte NC) pode, com as variáveis de sistema:

`$AA_OVR` Override axial

`$AC_OVR` Override de trajetória

ser lido e gravado em ações sincronizadas.

O Override especificado pelo PLC para ações sincronizadas pode, nas variáveis de sistema:

`$AA_PLC_OVR` Override axial

`$AC_PLC_OVR` Override de trajetória

é disponibilizado para leitura.

O Override resultante

para ações sincronizadas nas variáveis de sistema:

`$AA_TOTAL_OVR` Override axial

`$AC_TOTAL_OVR` Override de trajetória

é disponibilizado para leitura.

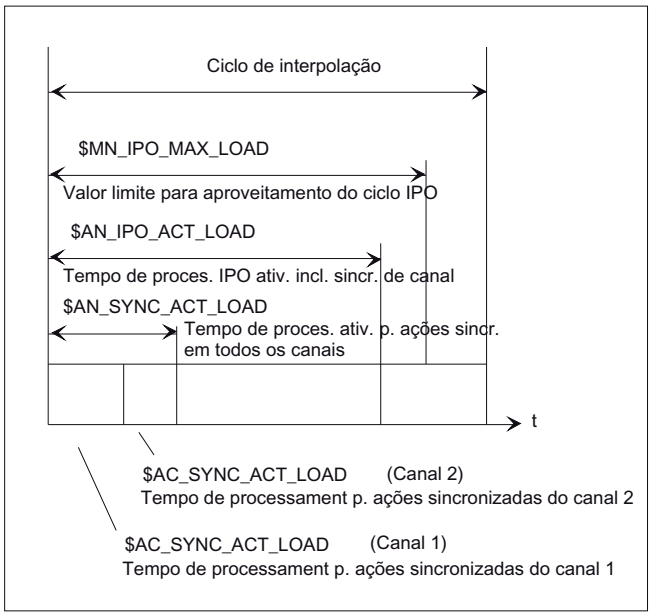
O override resultante é calculado como:

`$AA_OVR * $AA_PLC_OVR` ou
`$AC_OVR * $AC_PLC_OVR`

10.4.30 Avaliação do tempo usado das ações sincronizadas

Função

Em um ciclo de interpolação tanto devem ser interpretadas ações sincronizadas como calculados os movimentos pelo NC. Com as variáveis de sistema apresentadas a seguir, as ações sincronizadas são informadas sobre o tempo atual das ações sincronizadas no ciclo de interpolação e sobre o tempo de processamento do controlador de posição.


Ciclo de interpolação
$MN_IPO_MAX_LOAD
Valor limite para aproveitamento do ciclo IPO
$AN_IPO_ACT_LOAD
Tempo de proces. IPO ativ. incl. sincr. de canal
$AN_SYNC_ACT_LOAD
Tempo de proces. ativ. p. ações sincr. em todos os canais
t
$AC_SYNC_ACT_LOAD (Canal 2)
Tempo de processament p. ações sincronizadas do canal 2
$AC_SYNC_ACT_LOAD (Canal 1)
Tempo de processament p. ações sincronizadas do canal 1

Significado

As variáveis somente possuem valores válidos se o dado de máquina `$MN_IPO_MAX_LOAD` for maior que 0. Caso contrário as variáveis sempre indicam o tempo de processamento bruto, tanto no SINUMERIK powerline como no solution line, onde não se considera mais as interrupções geradas através da HMI. O tempo de processamento bruto resulta do(a):

- tempo da ação sincronizada,
- tempo do controle de posição e do
- tempo de processamento restante da interpolação sem interrupções condicionadas de HMI

As variáveis de sistema sempre contém os valores do ciclo de interpolação **anterior**.

$AN_IPO_ACT_LOAD	atual tempo de processamento de interpolação (incl. ações sincronizadas de todos canais)
$AN_IPO_MAX_LOAD	maior tempo de processamento da interpolação (incl. ações sincronizadas de todos canais)
$AN_IPO_MIN_LOAD	menor tempo de processamento da interpolação (incl. ações sincronizadas de todos canais)
$AN_IPO_LOAD_PERCENT	atual tempo de processamento de interpolação em relação ao ciclo de interpolação (%).
$AN_SYNC_ACT_LOAD	atual tempo de processamento para ações sincronizadas de todos canais
$AN_SYNC_MAX_LOAD	maior tempo de processamento para ações sincronizadas de todos canais
$AN_SYNC_TO_IPO	porcentagem de todas ações sincronizadas no tempo total de processamento da interpolação (de todos canais)
$AC_SYNC_ACT_LOAD	atual tempo de processamento para ações sincronizadas no canal
$AC_SYNC_MAX_LOAD	maior tempo de processamento para ações sincronizadas no canal
$AC_SYNC_AVERAGE_LOAD	tempo médio de processamento para ações sincronizadas no canal
$AN_SERVO_ACT_LOAD	atual tempo de processamento do controlador de posição
$AN_SERVO_MAX_LOAD	maior tempo de processamento do controlador de posição
$AN_SERVO_MIN_LOAD	menor tempo de processamento do controlador de posição

Variável do comunicado de sobrecarga:

Através do dado de máquina $MN_IPO_MAX_LOAD se ajusta, a partir de qual tempo de processamento bruto da interpolação (em % do ciclo IPO) a variável de sistema $AN_IPO_LOAD_LIMIT deve ser passada para TRUE. Se a atual carga estiver novamente abaixo deste limite, então a variável passa novamente para FALSE. Se o dado de máquina for 0, então a função de diagnóstico inteira é desativada.

Através da avaliação do $AN_IPO_LOAD_LIMIT o usuário pode definir sua própria estratégia para evitar um excesso de níveis.

10.5 Ciclos tecnológicos

Função

Como ação em ações sincronizadas também se pode chamar programas, mas que somente podem ser formados por funções permitidas como ações em ações sincronizadas. Os programas formados desta forma são chamados de ciclos de tecnologia.

Os ciclos de tecnologia são armazenados como subrotinas no comando.

Em um canal podem ser processados paralelamente vários ciclos de tecnologia ou ações.

Programação

Para programação de ciclos de tecnologia são aplicadas as seguintes regras:

- O fim de programa é programado com M02/M17/M30/RET.
- Dentro de um nível de programa, todas as ações indicadas no ICYCOF, podem ser executadas sem ciclos de espera em um ciclo.
- Podem ser consultados sucessivamente até 8 ciclos de tecnologia por ação sincronizada.
- Os ciclos de tecnologia também são possíveis em ações sincronizadas ativadas por blocos.
- Podem ser programadas tanto estruturas de controle IF como instruções de salto GOTO, GOTOF e GOTOB.
- Para blocos com instruções DEF e DEFINE:
  - Instruções DEF e DEFINE são ignoradas na leitura de ciclos de tecnologia.
  - Elas geram alarmes nos casos de sintaxes incorretas ou incompletas.
  - Elas podem, sem serem criadas automaticamente, serem ignoradas da leitura sem gerar alarme.
  - Com atribuição de valores elas são totalmente consideradas como ciclo de programa de peça.

Transferência de parâmetros

Uma transferência de parâmetros para ciclos de tecnologia não é possível. São considerados tanto os tipos de dados simples, que são transferidos como parâmetros formais "Call by Value", como ajustes padrão, que são ativados na chamada dos ciclos de tecnologia. São eles:

- Valores padrão programados quando nenhum parâmetro de transferência estiver programado.
- Parâmetros padrão com valores iniciais.
- Parâmetros atuais não inicializados transferidos com um valor padrão.

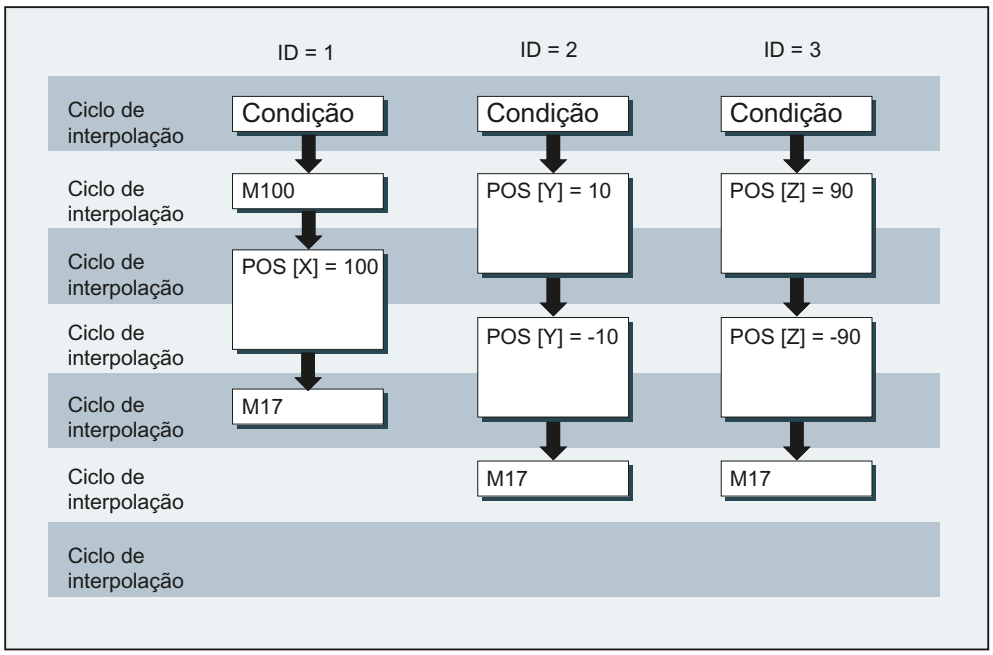
Seqüência de operação

Os ciclos de tecnologia são iniciados assim que suas condições forem preenchidas. Cada linha de um ciclo de tecnologia é executado em um ciclo de interpolação separado. Para eixos de posicionamento são necessários vários ciclos de interpolação para execução. Outras funções são executadas em um ciclo. No ciclo de tecnologia a execução dos blocos é seqüencial.

Se no mesmo ciclo de interpolação forem chamadas ações que se excluem mutuamente, então é ativada a ação que for chamada pela ação sincronizada de número ID maior.

Exemplos

Exemplo 1: Programas de eixos são iniciados com a definição de entradas digitais



Programa principal:

```
Código de programa                    Comentário
ID=1 EVERY $A_IN[1]==1 DO EIXO_X  ;  Se a entrada 1 está em 1, inicia o programa de eixo EIXO_X.
ID=2 EVERY $A_IN[2]==1 DO EIXO_Y  ;  Se a entrada 2 está em 1, inicia o programa de eixo EIXO_Y.
ID=3 EVERY $A_IN[3]==1 DO EIXO_Z  ;  Se a entrada 3 está em 1, inicia o programa de eixo EIXO_Z.
M30
```

Programa de eixo EIXO_X:

```
Código de programa
M100
POS[X]=100 FA[X]=300
M17
```

Programa de eixo EIXO_Y:

```
Código de programa
POS[Y]=10 FA[Y]=200
POS[Y]=-10
M17
```

Programa de eixo EIXO_Z:

```
Código de programa
POS[Z]=90 FA[Z]=250
POS[Z]=-90
M17
```

Exemplo 2: Diferentes seqüências de programa no ciclo de tecnologia

```
Código de programa
PROC CYCLE
N10 DEF REAL VALOR=12.3
N15 DEFINE ABC AS G01
```

Os dois blocos são ignorados na leitura sem gerar alarme e sem criar as variáveis ou macros.

```
Código de programa
PROC CYCLE
N10 DEF REAL
N15 DEFINE ABC G01
```

Os dois blocos continuam a gerar o alarme NC, pois a sintaxe não está corretamente escrita.

```
Código de programa
PROC CYCLE
N10 DEF AXIS EIXO1=XX2
```

Se o eixo XX2 não for conhecido, será emitido o alarme 12080. Caso contrário o bloco será ignorado na leitura sem gerar alarme e sem criar variáveis.

```
Código de programa
PROC CYCLE
N10 DEF AXIS EIXO1
N15 G01 X100 F1000
N20 DEF REAL VALOR1
```

O bloco `N20` sempre gera o alarme 14500, pois após o 1º comando de programa não pode haver nenhuma instrução de definição.

10.5.1 Variável de contexto ($P_TECCYCLE)

Função

Com a ajuda da variável $P_TECCYCLE os programas podem ser divididos em programas de ações sincronizadas e programas de pré-processamento. Desse modo, como alternativa, é possível executar os blocos escritos com sintaxe correta ou as seqüências de programa como se fossem ciclo de programa de peça.

Interpretação de variáveis de contexto

A variável de contexto $P_TECCYCLE permite controlar as interpretações específicas de contexto de partes de programa em ciclos de tecnologia:

```
IF $P_TECCYCLE==TRUE
...                          ; Seqüência de programa para ciclo de tecnologia na ação
                               síncrona.
ELSE
...                          ; Seqüência de programa para ciclo de programa de peça.
ENDIF
```

Indicação

Um bloco com sintaxe de programa incorreta ou não permitida, assim como as atribuições de valores não conhecidas, também geram um alarme no ciclo de programa de peça.

Exemplo

Seqüência de programa com consulta do $P_TECCYCLE no ciclo de tecnologia:

```
Código de programa              Comentário
PROC CYCLE
N10 DEF REAL VALOR1             ; Será ignorado na leitura do ciclo de tecnologia.
N15 G01 X100 F1000
N20 IF $P_TECCYCLE==TRUE
...                             ; Seqüência de programa para ciclo de tecnologia (sem variável VALOR1).
N30 ELSE
...                             ; Seqüência de programa para ciclo de programa de peça (a variável
                                  VALOR1 está presente).
N40 ENDIF
```

10.5.2 Parâmetro Call-by-Value

Função

Os ciclos de tecnologia podem ser definidos com parâmetros Call-by-Value. Como parâmetros são possíveis tipos de dados simples como INT, REAL, CHAR, DTRING, AXIS e BOOL.

Indicação

Os parâmetros formais, que transferem Call-by-Value, não podem ser campos.

Os parâmetros atuais também podem ser compostos por parâmetros Default (veja "Inicialização de parâmetros Default (Página 642)").

Sintaxe

```
ID=1 WHEN $AA_IW[X]>50 DO TEC(IVAL,RVAL,,SVAL,AVAL)
```

No caso dos parâmetros atuais não inicializados são transferidos valores Default:

```
ID=1 WHEN $AA_IW[X]>50 DO TEC(IVAL,RVAL,,SYG_SS[0],AVAL)
```

10.5.3 Inicialização de parâmetros Default

Função

Os parâmetros Default também podem ser atribuídos com um valor inicial na instrução PROC.

Sintaxe

Atribuir parâmetros Default no ciclo de tecnologia:

```
PROC TEC (INT IVAL=1, REAL RVAL=1.0, CHAR CVAL='A', STRING[10] SVAL="ABC", AXIS
AVAL=X, BOOL BVAL=TRUE)
```

Se um parâmetro atual é composto por um parâmetro Default, será transferido o valor inicial da instrução PROC. Isto se aplica tanto no programa de peça como em ações sincronizadas.

Exemplo

Código de programa	Comentário
`TEC (IVAL, RVAL, SVAL, AVAL)`	`;  com CVAL e BVAL vale o valor inicial`

10.5.4 Controle da execução de ciclos de tecnologia (ICYCOF, ICYCON)

Função

Para controle da execução cronológica de ciclos de tecnologia servem os comandos de linguagem ICYCOF e ICYCON.

Com ICYCOF todos os blocos de um ciclo de tecnologia são executados somente em um ciclo de interpolação. Todas ações, cuja execução requer vários ciclos, resultam em processos paralelos de usinagem com ICYCOF.

Aplicação

Com ICYCON os movimentos de eixos de comando podem fazer com que a execução de um ciclo de tecnologia seja prorrogado. Se isto não for desejado, com ICYCOF todas as ações podem ser executadas sem tempos de espera em um ciclo de interpolação apenas.

Sintaxe

Para a execução cíclica de ciclos de tecnologia se aplica:

`ICYCON` Todo bloco de um ciclo de tecnologia é executado depois do ICYCON em um ciclo separado de interpolação

`ICYCOF` Todos blocos seguintes de um ciclo de tecnologia são executados após o ICYCOF em um ciclo de interpolação apenas

Indicação

Os dois comandos de linguagem ICYCON e ICYCOF somente atuam dentro do nível do programa. No programa de peça os dois comandos são simplesmente ignorados na leitura, sem reação nenhuma.

Exemplo para modo de execução ICYCOF

Código de programa	**Comentário**
Ciclo de interpolação	; PROC TECHNOCYC
1.	; $R1=1
2.25	; POS[X]=100
26.	; ICYCOF
26.	; $R1=2
26.	; $R2=$R1+1
26.	; POS[X]=110
26.	; $R3=3
26.	; RET

10.5.5 Concatenação de ciclos de tecnologia

Função

Podem ser executados até 8 ciclos de tecnologia ligados em série. Desse modo é possível se programar vários ciclos de tecnologia em uma ação sincronizada.

Sintaxe

```
ID=1 WHEN $AA_IW[X]>50 DO TEC1($R1) TEC2 TEC3(X)
```

Seqüência de usinagem

Os ciclos de tecnologia são executados em série (concatenados) da esquerda para direita, de acordo com a programação especificada. Se um ciclo deve ser executado em modo ICYCON, então este retardará todos processamentos seguintes. Um alarme ocorrente cancela todas ações seguintes.

10.5.6 Ciclos de tecnologia em ações sincronizadas por blocos

Função

Os ciclos de tecnologia também são possíveis em ações sincronizadas por blocos.

Se o tempo de execução de um ciclo de tecnologia for mais longo que o tempo de execução do respectivo bloco, então o ciclo de tecnologia é cancelado na mudança de blocos.

Indicação

Um ciclo de tecnologia não impede a mudança de blocos.

10.5.7 Estruturas de controle (IF)

Função

Para derivações na seqüência de execução dos ciclos de tecnologia podem ser utilizadas estruturas de controle IF nas ações sincronizadas.

Sintaxe

```
IF <condição>
$R1=1
[ELSE] opcional
$R1=0
ENDIF
```

10.5.8 Instruções de salto (GOTO, GOTOF, GOTOB)

Função

Nos ciclos de tecnologia são possíveis as instruções de salto GOTO, GOTOF e GOTOB. Os Labels indicados devem estar presentes na subrotina, para que não seja emitido nenhum alarme.

Indicação

Labels e números de blocos somente podem ser constantes.

Sintaxe

Saltos incondicionais

`GOTO` Label, número de bloco

`GOTOF` Label, número de bloco

`GOTOB` Label, número de bloco

Instruções de salto e destinos de salto

`GOTO`	Salte primeiro para frente, depois para trás.
`GOTOF`	Salte para frente
`GOTOB`	Salte para trás
`Label:`	Marcador de salto
`Número do bloco`	Destino de salto para este bloco
`N100`	O número de bloco é bloco secundário
`:100`	O número de bloco é bloco principal

10.5.9 Bloquear, liberar, resetar (LOCK, UNLOCK, RESET)

Função

A execução de um ciclo de tecnologia pode ser bloqueada, liberada e resetada através de outra ação sincronizada modal.

Sintaxe

```
LOCK(<n1>,<n2>,...)
UNLOCK(<n1>,<n2>,...)
RESET(<n1>,<n2>,...)
```

Significado

`LOCK`	Comando para bloqueio de ações sincronizadas A ação ativa é interrompida.
`UNLOCK`	Comando para liberação de ações sincronizadas
`RESET`	Comando para resetamento de ciclos de tecnologia
`<n1>,<n2>,...`	Números de identificação das ações sincronizadas e ciclos de tecnologia que devem ser bloqueados, liberados ou resetados.

Travamento de ações sincronizadas

As ações sincronizadas de números ID **<n> = 1 ... 64** podem ser bloqueadas pelo PLC. Desse modo a respectiva condição não será mais avaliada e a execução da respectiva função será bloqueada no NCK.

Com um sinal da interface do PLC podem ser bloqueadas todas ações sincronizadas globalmente.

Indicação

Como padrão, uma ação sincronizada programada está ativa e pode ser protegida contra gravação/bloqueio através de dado de máquina.

As ações sincronizadas definidas pelo fabricante da máquina não deveriam ser influenciadas pelo cliente final.

Exemplos

Exemplo 1: Bloquear ações sincronizadas (LOCK)

```
Código de programa
N100 ID=1 WHENEVER $A_IN[1]==1 DO M130
...
N200 ID=2 WHENEVER $A_IN[2]==1 DO LOCK(1)
```

Exemplo 2: Liberar ações sincronizadas (UNLOCK)

```
Código de programa
N100 ID=1 WHENEVER $A_IN[1]==1 DO M130
...
N200 ID=2 WHENEVER $A_IN[2]==1 DO LOCK(1)
...
N250 ID=3 WHENEVER $A_IN[3]==1 DO UNLOCK(1)
```

Exemplo 3: Interromper ciclo de tecnologia (RESET)

```
Código de programa
N100 ID=1 WHENEVER $A_IN[1]==1 DO M130
...
N200 ID=2 WHENEVER $A_IN[2]==1 DO RESET(1)
```

10.6 Cancelar ação sincronizada (CANCEL)

Função

Com o comando CANCEL uma ação síncrona ativa pode ser cancelada (deletada) modal ou estaticamente do programa de peça.

Se for cancelada uma ação sincronizada e, enquanto isto o movimento do eixo de posicionamento que foi ativado pela mesma ainda estiver ativo, então este movimento do eixo de posicionamento será finalizado. Se isto não for desejado, o movimento de eixo pode ser desacelerado com a anulação do curso restante antes do comando CANCEL.

Sintaxe

```
CANCEL(<n1>,<n2>,...)
```

Significado

CANCEL:	Comando para cancelamento de ações sincronizadas programadas
<n1>,<n2>,...:	Números de identificação das ações sincronizadas que devem ser canceladas **Nota:** Sem a indicação dos números de identificação **todas** ações síncronas são canceladas de forma modal/estática.

Exemplos

Exemplo 1: Cancelamento de ação síncrona

Código de programa	Comentário
N100 ID=2 WHENEVER $A_IN[1]==1 DO M130	
...	
N200 CANCEL(2)	; Cancela a ação síncrona modal nº 2.

Exemplo 2: Anulação do curso restante antes do cancelamento da ação síncrona

Código de programa	Comentário
N100 ID=17 EVERY $A_IN[3]==1 DO POS[X]=15 FA[X]=1500	; Partida do movimento do eixo de posicionamento.
...	
N190 WHEN ... DO DELDTG(X)	; Finalização do movimento do eixo de posicionamento.
N200 CANCEL(17)	; Cancela a ação síncrona modal nº 17.

10.7 Comportamento de comando em determinados estados operacionais

POWER ON

Em princípio nenhuma ação síncrona está ativa com POWER ON. As ações sincronizadas estáticas podem ser ativadas imediatamente com Power On por uma subrotina assíncrona (`ASUP`) iniciada pelo PLC.

Mudança do modo de operação

Com a palavra-chave `IDS` as ações síncronas ativas permanecem ativas depois de uma mudança do modo de operação. Todas demais ações síncronas tornam-se inativas com a mudança do modo de operação (p. ex. posicionamento de eixo) e são novamente ativadas com o reposicionamento e o retorno ao modo de operação automático.

RESET

Com NC-RESET todas são encerradas através de ações síncronas ativas por blocos e modalmente. As ações sincronizadas estáticas permanecem ativas. A partir destas podem ser iniciadas novas ações. Se um movimento de eixo de comando estiver ativo no RESET, então este será cancelado. As ações síncronas finalizadas do tipo WHEN não são mais executadas após o RESET.

Comportamento após RESET		
Ação sincronizada/ Ciclo de tecnologia	**modal/por bloco**	**estático (IDS)**
	A ação ativa é cancelada, as ações síncronas são canceladas	A ação ativa é cancelada, o ciclo de tecnologia é resetado.
Eixo/ fuso a ser posicionado	O movimento é cancelado.	O movimento é cancelado.
fuso de rotação controlada	$MA_SPIND_ACTIVE_AFTER_RESET==1: O fuso permanece ativo $MA_SPIND_ACTIVE_AFTER_RESET==0: Fuso pára.	
Acoplamento de valor mestre	$MC_RESET_MODE_MASK, Bit13 == 1: O acoplamento de valor mestre permanece ativo $MC_RESET_MODE_MASK, Bit13 == 0: O acoplamento de valor mestre é desfeito	
Processos de medição	Os processos de medição iniciados a partir de ações síncronas são cancelados.	Os processos de medição iniciados a partir de ações síncronas estáticas são cancelados.

NC-Stop

Ações sincronizadas **estáticas** permanecem ativas com NC-Stop. Os movimentos iniciados a partir de ações sincronizadas estáticas não são cancelados. As ações sincronizadas **locais de programa** que pertencem ao bloco ativo permanecem ativas, os movimentos resultantes destas serão cancelados.

Fim do programa

O fim do programa e a ação sincronizada não tem influência um sobre o outro. As atuais ações sincronizadas também são encerradas após o fim do programa. As ações sincronizadas ativas no bloco M30 permanecem ativas no bloco M30. Se desejado, a ação síncrona com `CANCEL` deve ser cancelada antes do fim do programa.

Comportamento após o fim do programa		
Ação sincronizada/ Ciclo de tecnologia	**modais / por bloco → são cancelados**	**estáticos (IDS) → são mantidos**
Eixo/ fuso a ser posicionado	O M30 é prorrogado até o eixo/fuso estar parado.	O movimento continua.
fuso de rotação controlada	Fim do programa: $MA_SPIND_ACTIVE_AFTER_RESET==1: O fuso permanece ativo $MA_SPIND_ACTIVE_AFTER_RESET==0: O fuso pára O fuso permanece ativo com a mudança do modo de operação.	O fuso permanece ativo.
Acoplamento de valor mestre	$MC_RESET_MODE_MASK, Bit13 == 1: O acoplamento de valor mestre permanece ativo $MC_RESET_MODE_MASK, Bit13 == 0: O acoplamento de valor mestre é desfeito	O acoplamento iniciado a partir da ação síncrona estática, permanece.
Processos de medição	Os processos de medição iniciados a partir de ações síncronas são cancelados.	Os processos de medição iniciados a partir de ações síncronas estáticas permanecem ativos.

Localização de blocos

As ações sincronizadas encontradas durante a localização de blocos são coletadas e avaliadas no NC-Start, as ações pertencentes são iniciadas da mesma forma. As ações sincronizadas estáticas também tem efeito durante a localização de blocos. Se durante a localização de blocos forem encontrados coeficientes de polinômio programados com `FCTDEF`, estes serão ativados diretamente.

Interrupção de programa através de subrotina assíncrona ASUP

Início da ASUP:

As ações sincronizadas de movimentos estáticas e modais são mantidas e também estão ativas na subrotina assíncrona.

Fim da ASUP:

Se a subrotina assíncrona não for continuada com REPOS, então as ações síncronas de movimentos modais e estáticas que foram modificadas na subrotina assíncrona continuam a ter efeito no programa principal.

Reposicionamento (REPOS)

Após o reposicionamento (REPOS) as ações sincronizadas que estavam ativas no bloco interrompido são novamente ativadas. As ações síncronas modais alteradas a partir da subrotina assíncrona não estão mais ativas na execução do bloco restante após o REPOS.

Os coeficientes de polinômio programados com `FCTDEF` não são influenciados pelas subrotinas assíncronas e REPOS. Independentemente de como foram programados, eles podem ser aplicados em qualquer momento na subrotina assíncrona e no programa principal, mesmo após a execução do REPOS.

Comportamento em alarmes

Os movimentos de eixo e de fuso iniciados através de ações síncronas são desacelerados, quando um alarme com parada de movimento estiver ativo. Todas demais ações (como definição de saída) continuam sendo executadas.

Quando a própria ação síncrona dispara um alarme, então ocorre um cancelamento de processamento e as ações seguintes desta ação síncrona não serão mais executadas. Se a ação síncrona estiver ativa modalmente, ela não será mais processada no próximo ciclo de interpolação. O alarme somente será emitido uma vez. Todas demais ações síncronas continuam seu processamento.

Os alarmes que tiverem a parada de interpretação como reação de alarme, apenas terão efeito após a execução dos blocos já decodificados.

Se um ciclo de tecnologia dispara um alarme com parada de movimento, então o ciclo de tecnologia não será mais processado.

Oscilação 11

11.1 Oscilação assíncrona (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB)

Função

Um eixo oscilante alterna seu movimento entre os dois pontos de reversão 1 e 2 com o avanço especificado, até o movimento alternado ser desativado.

Os demais eixos podem ser interpolados quantas vezes for necessário durante o movimento alternado. Uma penetração contínua é alcançada através de um movimento de percurso ou com um eixo de posicionamento. Entretanto, neste caso não existe **nenhuma relação** entre os movimentos alternado e de penetração.

Propriedades da oscilação assíncrona

- A oscilação assíncrona está ativa em função dos eixos, além dos limites de bloco.
- Através do programa de peça se garante uma ativação do movimento alternado sincronizada com os blocos.
- Uma interpolação comum de vários eixos e uma sobreposição de trechos de oscilação não são possíveis.

Programação

Através dos seguintes comandos é possível fazer uma ativação e controle da oscilação assíncrona durante a execução do programa NC a partir do programa de peça.

Os valores programados são registrados nos respectivos dados de ajustes de modo sincronizadamente aos blocos no processamento principal e permanecem ativos até a próxima alteração.

Sintaxe

```
OSP1[<eixo>]=<valor> OSP2[<eixo>]=<valor>
OST1[<eixo>]=<valor> OST2[<eixo>]=<valor>
FA[<eixo>]=<valor>
OSCTRL[<eixo>]=(<opção de definição>,<opção de resetamento>)
OSNSC[<eixo>]=<valor>
OSE[<eixo>]=<valor>
OSB[<eixo>]=<valor>
OS[<eixo>]=1
OS[<eixo>]=0
```

Significado

`<eixo>` Nome do eixo oscilante

`OS` Ativação/desativação da oscilação

Valor: 1 **A**tivação da oscilação

0 **Desa**tivação da oscilação

`OSP1` Definição da posição do ponto de reversão 1

`OSP2` Definição da posição do ponto de reversão 2

Nota:
Se nenhum deslocamento incremental estiver ativo, então a posição incremental será calculada para a última posição de reversão correspondente programada no programa NC.

`OST1` Definição de tempo de parada no ponto de reversão 1 em [s]

`OST2` Definição de tempo de parada no ponto de reversão 2 em [s]

`<valor>`:
- -2 A interpolação é continuada sem espera pela parada exata
- -1 Espera pela parada exata aproximada
- 0 Espera pela parada exata fina
- >0 Espera pela parada exata fina e em seguida espera pelo tempo de parada especificado

  Nota:
  A unidade do tempo de espera é idêntica à do tempo de parada programado com `G4`.

`FA` Definição de velocidade de avanço

Como velocidade de avanço se aplica a que foi definida para o eixo de posicionamento. Se nenhuma velocidade de avanço foi definida, será aplicado o valor armazenado no dado de máquina.

OSCTRL Indicação de opções de definição e de resetamento

Os valores de opção 0 - 3 classificam o comportamento nos pontos de reversão na desativação. Pode ser selecionada uma das variantes 0 - 3. As demais configurações podem ser combinadas conforme necessidade com a variante selecionada. Várias opções são anexadas através de sinais positivos (+).

<valor>:

0 Ao ser desativado o movimento oscilante, a parada ocorre no próximo ponto de reversão (pré-ajuste)

Nota:
Possível somente com o resetamento dos valores 1 e 2.

1 Quando o movimento alternado é desativado, parar no ponto de reversão 1

2 Quando o movimento alternado é desativado, parar no ponto de reversão 2

3 Quando o movimento alternado é desativado, não aproximar nenhum ponto de reversão, isto se nenhum curso de passada final foi programado

4 Após a passada final aproximar a posição final

8 Se o movimento oscilante for cancelado através da anulação do curso restante, os cursos de passada final devem ser executados em seguida e, se necessário, executada a aproximação da posição final.

16 Se o movimento oscilante é cancelado através da anulação do curso restante, a posição de reversão correspondente deve ser aproximada como no caso da desativação.

32 O avanço alterado só estará ativo a partir do próximo ponto de reversão

64 FA igual a 0, FA = 0: A sobreposição de cursos está ativa

FA igual a 0, FA <> 0: A sobreposição de velocidade está ativa

128 Para eixo rotativo DC (curso mais curto)

256 O curso de passada final é executado como curso duplo.(padrão) 1=curso de passada final é executado como curso simples.

512 Primeiro aproximar a posição de partida

`OSNSC` Definição do número de cursos de passada final

`OSE` Definição da posição final (no WCS), que deve ser aproximada depois de ser desativado o movimento oscilante

Nota:
Com a programação do `OSE` é ativada de forma implícita a opção 4 para o `OSCTRL`.

`OSB` Definição da posição de partida (no WCS), que deve ser aproximada antes de ser ativado o movimento oscilante

A posição de partida é aproximada antes do ponto de reversão 1. Quando a posição de partida coincide com a posição de reversão 1, então a aproximação ocorre na próxima posição de reversão 2. Ao ser alcançada a posição de partida nenhum tempo de parada entra em ação, mesmo se a posição de partida coincidir com a posição de reversão 1, ao invés disso, espera-se a parada exata fina. Uma condição de parada exata ajustada será mantida.

Nota:
Para que a posição de partida seja aproximada, no dado de ajuste SD43770 $SA_OSCILL_CTRL_MASK deve estar definido o Bit 9.

Exemplos

Exemplo 1: O eixo oscilante deve oscilar entre dois pontos de reversão

O eixo oscilante Z deve oscilar entre as posições 10 e 100. O ponto de reversão 1 deve ser aproximado com parada exata fina, e o ponto de reversão 2 com parada exata aproximada. O avanço para o eixo oscilante deve ser 250. No final da usinagem devem ser executados 3 cursos de passada final e o eixo oscilante deve aproximar-se da posição final 200. O avanço para o eixo de penetração deve ser 1, o fim da penetração no sentido X deve ser alcançado na posição 15.

```
Código de programa                  Comentário
WAITP(X,Y,Z)                        ; Posição de saída.
G0 X100 Y100 Z100                   ; Comutação para o modo de eixo de posicionamento.
WAITP(X,Z)
OSP1[Z]=10 OSP2[Z]=100              ; Ponto de reversão 1, ponto de reversão 2.
OSE[Z]=200                          ; Posição final.
OST1[Z]=0 OST2[Z]=-1                ; Tempo de parada no U1: Parada exata fina
                                    ; Tempo de parada no U2: Parada exata aproximada
FA[Z]=250 FA[X]=1                   ; Avanço do eixo oscilante, avanço do eixo de
                                      penetração.
OSCTRL[Z]=(4,0)                     ; Opções de definição.
OSNSC[Z]=3                          ; 3 cursos de passada final.
OS[Z]=1                             ; Partida da oscilação.
WHEN $A_IN[3]==TRUE DO DELDTG(X)    ; Anulação de curso restante.
POS[X]=15                           ; Posição de saída do eixo X
POS[X]=50                             Posição final do eixo X.
OS[Z]=0                             ; Parada da oscilação.
M30
```

Indicação

A seqüência de comandos `OSP1[Z]=...` até `OSNCS[Z]=...` também pode ser programada em um bloco.

Exemplo 2: Oscilação com alteração Online da posição de reversão

Os dados de ajuste necessários para a oscilação assíncrona podem ser ajustados no programa de peça.

Se os dados de ajuste forem escritos diretamente no programa de peça, então a alteração estará ativa no momento do pré-processamento. O comportamento síncrono pode ser obtido com uma parada de pré-processamento (`STOPRE`).

```
Código de programa                                                    Comentário
$SA_OSCILL_REVERSE_POS1[Z]=-10
$SA_OSCILL_REVERSE_POS2[Z]=10
G0 X0 Z0
WAITP(Z)
ID=1 WHENEVER $AA_IM[Z] < $$AA_OSCILL_REVERSE_POS1[Z] DO $AA_OVR[X]=0     ; Se o valor real do eixo
ID=2 WHENEVER $AA_IM[Z] < $$AA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[X]=0       oscilante ultrapassou o
                                                                            ponto de reversão, então
                                                                            o eixo de penetração será
                                                                            parado.
OS[Z]=1 FA[X]=1000 POS[X]=40                                          ; Ativação da oscilação.
OS[Z]=0                                                               ; Desativação da oscilação.
M30
```

Outras informações

Eixo oscilante

Para o eixo oscilante aplica-se:

- Todo eixo pode ser utilizado como eixo oscilante.
- Vários eixos oscilantes podem estar ativos ao mesmo tempo (máximo: número de eixos de posicionamento).
- Para o eixo oscilante sempre está ativa a interpolação linear `G1`, independentemente do atual código G.

O eixo oscilante pode:

- ser eixo de entrada para a transformação dinâmica
- ser eixo guia para eixos Gantry e eixos acoplados
- no deslocamento:
  - sem limitação de solavancos (BRISK)

    ou
  - com limitação de solavancos (SOFT)

    ou
  - com curva característica de aceleração dobrada (como eixos de posicionamento)

Pontos de reversão da oscilação

Para definição das posições oscilantes devem ser observados os atuais deslocamentos:

- Especificação absoluta

  `OSP1[Z]=<valor>`

  Posição do ponto de reversão = Soma dos deslocamentos + valor programado
- Especificação relativa

  `OSP1[Z]=IC(<valor>)`

  Posição do ponto de reversão = Ponto de reversão 1 + valor programado

Exemplo:

Código de programa

```
N10 OSP1[Z] = 100 OSP2[Z] = 110
...
...
N40 OSP1[Z] = IC(3)
```

WAITP

Se a oscilação deve ser realizada com um eixo geométrico, então esta deve ser liberada com WAITP para oscilação.

Com WAITP, depois de finalizada a oscilação, o eixo oscilante será registrado novamente como eixo de posicionamento e poderá ser utilizado normalmente.

Oscilação com ações síncronas de movimento e tempos de parada

Depois de expirar os tempos de parada ajustados, é realizada uma mudança interna de blocos durante a oscilação (visível nos novos cursos dos eixos). Na mudança de blocos é verificada a função de desativação. Neste caso é definida a função de desativação, após o ajuste de controle da seqüência de movimento (OSCTRL). *Este comportamento de tempo pode ser controlado pelo Override de avanço.*

Talvez depois disso ainda seja executado um curso oscilante antes dos cursos de passada final serem iniciados ou da posição final ser aproximada. *Neste caso temos a impressão que o comportamento de desativação mudou. Mas este não é o caso.*

11.2 Oscilação controlada através de ações síncronas (OSCILL)

Função

Neste tipo de oscilação apenas é permitido um movimento de penetração nos pontos de reversão ou dentro de áreas de reversão definidas.

Dependendo da necessidade, o movimento da oscilação pode, durante a penetração,

- ser continuado ou
- parado até que a penetração seja totalmente executada.

Sintaxe

1. **Definição do parâmetro para a oscilação**
2. **Definição de ações síncronas de movimentos**
3. **Atribuição de eixos, definição da penetração**

Significado

`OSP1[<eixo oscilante>]=`	Posição do ponto de reversão 1
`OSP2[<eixo oscilante>]=`	Posição do ponto de reversão 2
`OST1[<eixo oscilante>]=`	Tempo de parada no ponto de reversão 1 em segundos
`OST2[<eixo oscilante>]=`	Tempo de parada no ponto de reversão 2 em segundos
`FA[<eixo oscilante>]=`	Avanço do eixo oscilante
`OSCTRL[<eixo oscilante>]=`	Opções de definição e de resetamento
`OSNSC[<eixo oscilante>]=`	Número de cursos de passada final
`OSE[<eixo oscilante>]=`	Posição final
`WAITP(<eixo oscilante>)`	Liberação do eixo para a oscilação

Atribuição de eixos, penetração

```
OSCILL[<eixo oscilante>]=(<eixo de penetração 1>,<eixo de
penetração 2>,<eixo de penetração 3>)
POSP[<eixo de penetração>]=(<posição final>,<comprimento
parcial>,<modo>)
```

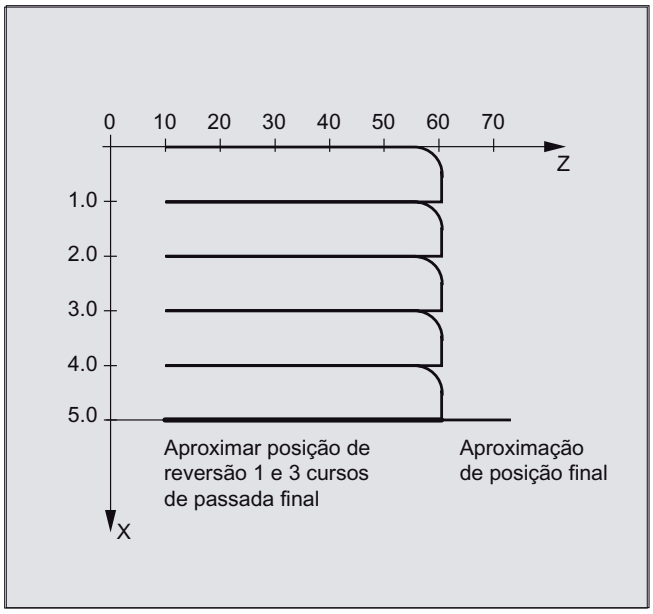
`OSCILL`:	Atribuir eixo(s) de penetração para eixo oscilante
`POSP`:	Definição das penetrações totais e parciais (veja capítulo Gerenciamento de arquivos e programas)
`Posição final`:	Posição final do eixo de penetração, depois de todas penetrações parciais serem executadas.
`Comprimento parcial`:	Tamanho da penetração parcial no ponto/área de reversão
`Modo`:	Divisão da penetração total em penetrações parciais = dois passos restantes de mesmo tamanho (pré-ajuste); = todas as penetrações parciais de mesmo tamanho

Ações sincronizadas de movimentos

`WHEN… … DO`	quando…, então…
`WHENEVER … DO`	sempre que…, então…

Exemplo

No ponto de reversão 1 não deve ser realizada nenhuma penetração. Para o ponto de reversão 2 a penetração deve ser realizada na distância ii2 antes do ponto de reversão 2 e o eixo oscilante não deve esperar no ponto de reversão pela finalização da penetração parcial. O eixo Z é o eixo oscilante e o eixo X é o eixo de penetração.



1. Parâmetro para a oscilação

Código de programa	Comentário
`DEF INT ii2`	`; Definição da variável para a área de reversão 2`
`OSP1[Z]=10 OSP2[Z]=60`	`; Definição do ponto de reversão 1 e 2`
`OST1[Z]=0 OST2[Z]=0`	`; Ponto de reversão 1: Parada exata fina`
	`Ponto de reversão 2: Parada exata fina`
`FA[Z]=150 FA[X]=0.5`	`; Avanço do eixo oscilante Z, avanço do eixo de penetração X`
`OSCTRL[Z]=(2+8+16,1)`	`; Desativação do movimento alternado no ponto de reversão 2; após o RWL realização da passada final e aproximação da posição final; após o RWL aproximação da respectiva posição de reversão`
`OSNC[Z]=3`	`; Cursos de passada final`
`OSE[Z]=70`	`; Posição final = 70`
`ii2=2`	`; Ajuste da área de reversão`
`WAITP(Z)`	`; Permite a oscilação do eixo Z`

2. Ação sincronizada de movimentos

Código de programa		Comentário
WHENEVER $AA_IM[Z]<$SA_OSCILL_REVERSE_POS2[Z] DO -> $AA_OVR[X]=0 $AC_MARKER[0]=0	;	Sempre que a atual posição do eixo oscilante Z em MCS for menor que o início da área de reversão 2, então defina o Override axial do eixo de penetração X em 0% e o marcador de índice 0 com o valor 0.
WHENEVER $AA_IM[Z]>=$SA_OSCILL_REVERSE_POS2[Z] DO $AA_OVR[Z]=0	;	Sempre que a atual posição do eixo oscilante Z em MCS for maior ou igual à posição de reversão 2, então defina o Override axial do eixo oscilante Z em 0%.
WHENEVER $AA_DTEPW[X]==0 DO $AC_MARKER[0] = 1	;	Sempre que o curso restante da penetração parcial for igual a 0, então defina o marcador de índice 0 com o valor 1.
WHENEVER $AC_MARKER[0]==1 DO $AA_OVR[X]=0 $AA_OVR[Z]=100	;	Sempre que o marcador de índice 0 for igual a 1, então defina o Override axial do eixo de penetração X em 0%. Com isso é evitada uma penetração muito antecipada (o eixo oscilante Z ainda não saiu da área de reversão 2, mas o eixo de penetração X está pronto para uma nova penetração). Passe o Override axial do eixo oscilante Z de 0% (ação da 2ª ação síncrona) para deslocamento novamente para 100%.

-> deve ser programado em um bloco

3. Início da oscilação

Código de programa		Comentário
OSCILL[Z]=(X) POSP[X]=(5,1,1)	;	Partida dos eixos Para o eixo oscilante Z é atribuído o eixo X como eixo ;de penetração. O eixo X deve movimentar-se até a posição final 5 em passos de 1.
M30	;	Fim do programa

Descrição

1. **Definir parâmetro de oscilação**
   Antes do bloco de movimento que contém tanto a atribuição do eixo de penetração e do eixo oscilante como a definição da penetração, devem ser definidos os parâmetros da oscilação (veja "Oscilação assíncrona").
2. **Definir ações sincronizadas de movimentos**
   Através das condições de sincronização ocorre:
   Supressão da penetração até que o eixo oscilante estiver dentro de uma área de reversão
   (ii1, ii2) ou em um ponto de reversão (U1, U2).
   Parar o **movimento alternado** no ponto de reversão durante a penetração.
   Reiniciar o **movimento alternado** depois de concluir a penetração parcial.
   Definir a **partida na próxima penetração parcial**.
3. **Atribuir eixo oscilante e eixo de penetração** assim como definir a **penetração total e parcial**.

Definir parâmetro de oscilação

Atribuição do eixo oscilante e eixo de penetração: OSCILL

```
OSCILL[eixo oscilante] = (eixo de penetração1, eixo de penetração2, eixo de penetração3)
```

Com o comando `OSCILL` são realizadas as atribuições de eixo e a partida do movimento alternado.

Podem ser atribuídos no máximo 3 eixos de penetração para um eixo oscilante.

Indicação

Antes da partida da oscilação devem ser definidas as condições de sincronização para a relação dos eixos.

Definir penetrações: POSP

```
POSP[eixo de penetração] = (Endpos, comprimento parcial, modo)
```

Com o comando `POSP` são comunicados ao comando:

- Penetração total (através da posição final)
- O tamanho da respectiva penetração parcial no ponto de reversão ou na área de reversão
- O comportamento da penetração parcial ao alcançar a posição final (através do modo)

Modo = 0	Para as duas últimas penetrações parciais é realizada uma divisão dos cursos restantes até o ponto de destino em dois passos restantes de mesmo tamanho (pré-ajuste).
Modo = 1	Todas penetrações parciais são de mesmo tamanho. Elas são calculadas a partir da penetração total.

Definir ações sincronizadas de movimentos

As ações sincronizadas de movimentos mencionadas a seguir são utilizadas de forma bem geral na oscilação.

Aqui encontramos exemplos para solução em determinadas solicitações que servirão como módulos para a criação de movimentos alternados específicos de usuário.

Indicação

Em casos específicos, as condições de sincronização também podem ser programadas de outra forma.

Palavras-chave

`WHEN … DO …`	quando…, então…
`WHENEVER … DO`	sempre que…, então…

Funções

Com os meios de linguagem descritos detalhadamente a seguir, as seguintes funções, podem ser

realizadas:

1. Penetração no ponto de reversão.
2. Penetração na área de reversão.
3. Penetração nos dois pontos de reversão.
4. Parada no movimento alternado no ponto de reversão.
5. Reiniciar o movimento alternado.
6. Não iniciar antecipadamente a penetração parcial.

Para todas ações sincronizadas aqui apresentadas como exemplo estão valendo as suposições:

- Ponto de reversão 1 < ponto de reversão 2
- Z = eixo oscilante
- X = eixo de penetração

Indicação

Para mais explanações, veja o capítulo Ações sincronizadas de movimentos.

Atribuir eixo oscilante e eixo de penetração assim como definir a penetração total e parcial

Penetração na área de reversão

O movimento de penetração deve iniciar dentro de uma área de reversão, antes do ponto de reversão ser alcançado.

As ações sincronizadas evitam o movimento de penetração até que o eixo oscilante entre na área de reversão.

Entre as suposições especificadas (veja acima) são obtidas as seguintes instruções:

Área de reversão 1:

```
WHENEVER
$AA_IM[Z]>$SA_OSCILL_RESE
RVE_POS1[Z]+ii1 DO
$AA_OVR[X] = 0
```

Sempre que a atual posição do eixo oscilante em MCS for maior que o início da área de reversão 1, então defina o Override axial do eixo de penetração em 0%.

Área de reversão 2:

```
WHENEVER
$AA_IM[Z]<$SA_OSCILL_RESE
RVE_POS2[Z]+ii2 DO
$AA_OVR[X] = 0
```

Sempre que a atual posição do eixo oscilante em MCS for menor que o início da área de reversão 2, então defina o Override axial do eixo de penetração em 0%.

Penetração no ponto de reversão

Não é executado nenhum movimento do eixo de penetração enquanto o eixo oscilante não alcançar o ponto de reversão.

Entre as suposições especificadas (veja acima) são obtidas as seguintes instruções:

Área de reversão 1:

```
WHENEVER
$AA_IM[Z]<>$SA_OSCILL_RES
ERVE_POS1[Z] DO
$AA_OVR[X] = 0 →
→ $AA_OVR[Z] = 100
```

Sempre que a atual posição do eixo oscilante Z em MCS for maior ou menor que a posição do ponto de reversão 1, então defina o Override axial do eixo de penetração X em 0% e o Override axial do eixo oscilante Z em 100%.

Área de reversão 2:

Para ponto de reversão 2:

```
WHENEVER
$AA_IM[Z]<>$SA_OSCILL_RES
ERVE_POS2[Z] DO
$AA_OVR[X] = 0 →
→ $AA_OVR[Z] = 100
```

Sempre que a atual posição do eixo oscilante Zu em MCS for maior ou menor que a posição do ponto de reversão 2, então defina o Override axial do eixo de penetração X em 0% e o Override axial do eixo oscilante Z em 100%.

Parada do movimento alternado no ponto de reversão

O eixo oscilante é parado no ponto de reversão, ao mesmo tempo é iniciado o movimento de penetração. O movimento alternado é continuado assim que o movimento de penetração for concluído.

Simultaneamente esta ação sincronizada pode ser utilizada para iniciar o movimento de penetração, caso este tenha sido parado por uma ação sincronizada anterior que ainda está ativa.

Entre as suposições especificadas (veja acima) são obtidas as seguintes instruções:

Área de reversão 1:

```
WHENEVER
$SA_IM[Z]==$SA_OSCILL_RES
ERVE_POS1[Z] DO
$AA_OVR[X] = 0 →
→ $AA_OVR[Z] = 100
```

Sempre que a atual posição do eixo oscilante em MCS for igual à posição de reversão 1, então defina o Override axial do eixo oscilante em 0% e o Override axial do eixo de penetração em 100%.

Área de reversão 2:

```
WHENEVER
$SA_IM[Z]==$SA_OSCILL_RES
ERVE_POS2[Z] DO
$AA_OVR[X] = 0 →
→ $AA_OVR[Z] = 100
```

Sempre que a atual posição do eixo oscilante Zu em MCS for igual à posição de reversão 2, então defina o Override axial do eixo oscilante X em 0% e o Override axial do eixo de penetração em 100%.

Avaliação Online do ponto de reversão

Se ao lado direito da comparação estiver uma variável de processamento principal marcada com `$$`, então as duas variáveis serão avaliadas durante o ciclo de interpolação e comparadas entre si.

Indicação

Para mais informações sobre este assunto, veja o capítulo "Ações sincronizadas de movimentos".

Reiniciar o movimento alternado

Esta ação sincronizada é utilizada para dar continuidade ao movimento do eixo oscilante quando o movimento de penetração parcial estiver concluído.

Entre as suposições especificadas (veja acima) são obtidas as seguintes instruções:

```
WHENEVER $AA_DTEPW[X]==0
DO $AA_OVR[Z] = 100
```

Sempre que o curso restante para a penetração parcial do eixo de penetração X em WCS for igual a zero, então defina o Override axial do eixo oscilante em 100%.

Próxima penetração parcial

Após a realização da penetração se deve evitar uma partida antecipada da próxima penetração parcial.

Para isso é utilizado um marcador específico de canal (`$AC_MARKER[índice]`) que é especificado no fim da penetração parcial (curso restante parcial = 0) e deletado depois de deixar a área de reversão. Então o próximo movimento de penetração é evitado com uma ação sincronizada.

Entre as suposições especificadas (veja acima) são obtidas as seguintes instruções, por exemplo, para o ponto de reversão 1:

1. Definir marcador:

```
WHENEVER $AA_DTEPW[X]==0
DO $AC_MARKER[1] = 1
```

Sempre que o curso restante da penetração parcial do eixo de penetração X em WS for igual a zero, então defina o marcador de índice 1 em 1.

2. Deletar marcador

```
WHENEVER $AA_IM[Z]<>
$SA_OSCILL_RESERVE_POS1[Z
] DO $AC_MARKER[1]= 0
```

Sempre que a atual posição do eixo oscilante Z em MCS for maior ou menor que a posição do ponto de reversão 1, então defina o marcador 1 em 0.

3. Evitar a penetração

```
WHENEVER $AC_MARKER[1]==1
DO $AA_OVR[X] = 0
```

Sempre que o marcador 1 for igual, então defina o Override axial do eixo de penetração X em 0%.

Estampagem e puncionamento 12

12.1 Ativação, desativação

12.1.1 Estampagem e puncionamento ativado ou desativado (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC)

Função

Ativação/desativação da estampagem e puncionamento

A função de estampagem e de puncionamento é ativada com PON e SON. O SPOF encerra todas funções específicas de estampagem e de puncionamento. Os comandos PON e SON ativados modalmente se encerram mutuamente, isto é, o PON desativa o SON e vice-versa.

Estampagem/puncionamento com guia

As funções SONS e PONS também ativam as funções de estampagem e de puncionamento.

Ao contrário do controle de curso no nível de interpolação ativo com SON/PON, para estas funções é realizado um controle baseado em sinais da inicialização do curso no nível servo. Com isso pode-se trabalhar com freqüências mais elevadas de curso, consequentemente com uma maior capacidade de estampagem.

Durante a avaliação de sinais na guia, todas funções que conduzem a uma alteração de posição dos eixos de puncionamento ou de estampagem (p. ex. manivela eletrônica, alterações de Frames via PLC, funções de medição) são bloqueadas.

Estampagem com retardo

O PDELAYON realiza a execução com retardo do curso de estampagem. O comando ativado modalmente tem função preparatória e normalmente está antes do PON. A estampagem é normalmente continuada após o PDELAYOF.

Indicação

O tempo de retardo é ajustado no dado de ajuste SD42400 $SC_PUNCH_DWELLTIME.

Aceleração em função do curso

Com PUNCHACC pode ser definida uma curva característica de aceleração, que define diferentes acelerações em função da distância entre os furos.

Segunda interface de estampagem

As máquinas que devem utilizar alternadamente uma segunda interface de estágio (segunda unidade de estampagem ou uma mídia equivalente), podem ser comutadas para um segundo par de entradas e saídas digitais rápidas do comando (par de I/O). A total funcionalidade de estampagem/puncionamento está disponível para as duas interfaces de estágio. A comutação entre a primeira e a segunda interface de estampagem é realizada através dos comandos `SPIF1` e `SPIF2`.

Indicação

Pré-requisito: Um segundo par de I/O precisa estar definido para a funcionalidade de estampagem através de dados de máquina (→ veja as informações do fabricante da máquina!).

Sintaxe

```
PON G... X... Y... Z...
SON G... X... Y... Z...
SONS G... X... Y... Z...
PONS G... X... Y... Z...
PDELAYON
PDELAYOF
PUNCHACC(<Smin>,<Amin>,<Smax>,<Amax>)
SPIF1/SPIF2
SPOF
```

Significado

`PON`	Ativação da estampagem
`SON`	Ativação do puncionamento
`PONS`	Ativação da estampagem com guia
`SONS`	Ativação do puncionamento com guia
`SPOF`	Desativação da estampagem/puncionamento
`PDELAYON`	Ativação da estampagem com retardo
`PDELAYOF`	Desativação da estampagem com retardo
`PUNCHACC`	Ativação da aceleração em função do curso
	Parâmetro:
	`<Smin>` Menor distância entre furos
	`<Amin>` Aceleração inicial
	A aceleração inicial `<Amin>` pode ser maior que o `<Amax>`.
	`<Smax>` Maior distância entre furos
	`<Amax>` Aceleração final
	`<Amax>` pode ser maior que o `<Amin>`.
`SPIF1`	Ativação da **primeira** interface de estampagem
	O controle de curso é realizado através do primeiro par de I/O rápido.

`SPIF2` Ativação da **segunda** interface de estampagem

O controle de curso é realizado através do segundo par de I/O rápido.

Nota:
Após o RESET ou na inicialização do comando é sempre a primeira interface de estampagem que está ativa. Se apenas uma interface de estampagem for usada, então esta programação não precisa ser feita.

Exemplos

Exemplo 1: Ativação do puncionamento

Código de programa	Comentário
...	
N70 X50 SPOF	; Posicionamento sem disparo de estampagem.
N80 X100 SON	; Ativação de puncionamento, disparo de um curso antes do movimento (X=50) e no fim do movimento programado (X=100).
...	

Exemplo 2: Estampagem com retardo

Código de programa	Comentário
...	
N170 PDELAYON X100 SPOF	; Posicionamento sem disparo de estampagem, ativação do disparo de estampagem com retardo.
N180 X800 PON	; Ativação da estampagem. Depois de ser alcançada a posição final, o curso de estampagem é executado com retardo.
N190 PDELAYOF X700	; Desativação da estampagem com retardo, disparo de estampagem normal no fim do movimento programado.
...	

Exemplo 3: Estampagem com duas interfaces de estágio

Código de programa	Comentário
...	
N170 SPIF1 X100 PON	; No fim do bloco ocorre um disparo de curso na primeira saída rápida. O sinal "Curso ativo" é monitorado na primeira entrada.
N180 X800 SPIF2	; O segundo disparo de curso ocorre na segunda saída rápida. O sinal "Curso ativo" é monitorado na segunda entrada.
N190 SPIF1 X700	; O controle de curso para os demais cursos é realizado com a primeira interface.
...	

Outras informações

Estampagem e puncionamento com guia (PONS/SONS)

A estampagem e puncionamento com guias simultâneos não são possíveis em vários canais. Somente o PONS ou o SONS pode estar ativo em um canal.

Aceleração em função do curso (PUNCHACC)

Exemplo:
```
PUNCHACC(2,50,10,100)
```

Distâncias entre fusos abaixo de 2mm:

O deslocamento é executado com uma aceleração de 50% da aceleração máxima.

As distâncias entre furos de 2mm até 10mm:

A aceleração é aumentada proporcionalmente à distância em 100%.

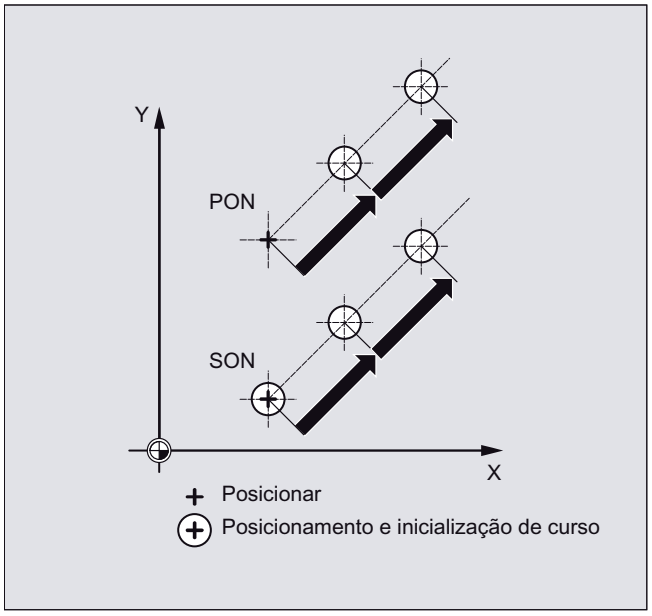
Distâncias entre furos maior que 10mm:

Deslocamento com uma aceleração de 100%.

Disparo do primeiro curso

O disparo do primeiro curso após a ativação da função é realizado em tempos diferentes para o puncionamento e estampagem:

- PON/PONS:
  - Todos cursos são realizados no fim do bloco, mesmo o primeiro bloco após a ativação.
- SON/SONS:
  - O primeiro curso após a ativação do puncionamento é realizado no início do bloco.
  - Todos demais cursos são ativados no fim do bloco.



Estampagem e puncionamento no local

Uma inicialização do curso somente ocorre quando o primeiro bloco contém uma informação de deslocamento para os eixos de estampagem ou puncionamento (eixos do plano ativo).

Mesmo assim, para disparar um curso no mesmo local, deve ser programado um dos eixos de estampagem/puncionamento com percurso 0.

Trabalhar com ferramentas rotativas

Indicação

Para empregar ferramentas rotativas tangencialmente ao percurso programado, utilize o controle tangencial.

Utilização de comandos M

Com a ajuda da tecnologia de macros ainda é possível utilizar funções M especiais ao invés dos comandos de linguagem (compatibilidade). Neste caso são aplicadas as seguintes correspondências com os sistemas antigos:

```
M20, M23    ≙  SPOF
M22         ≙  SON
M25         ≙  PON
M26         ≙  PDELAYON
```

Exemplo para arquivo de macro:

```
Código de programa                Comentário
DEFINE M25 AS PON                 ; Estampagem ativada
DEFINE M125 AS PONS               ; Estampagem com guia ativada
DEFINE M22 AS SON                 ; Puncionamento ativado
DEFINE M122 AS SONS               ; Puncionamento com guia ativado
DEFINE M26 AS PDELAYON            ; Estampagem com retardo ativada
DEFINE M20 AS SPOF                ; Estampagem, puncionamento desativado
DEFINE M23 AS SPOF                ; Estampagem, puncionamento desativado
```

Exemplo de programação:

```
Código de programa                Comentário
...
N100 X100 M20                     ; Posicionamento sem disparo de estampagem.
N110 X120 M22                     ; Ativação de puncionamento, disparo de curso
                                    antes e depois do movimento.
N120 X150 Y150 M25                ; Ativação de estampagem, disparo de curso no fim
                                    do movimento.
...
```

12.2 Preparação automática do curso

Função

Segmentação em trechos

Para a estampagem e puncionamento ativados, tanto o SPP como o SPN realizam uma segmentação do trecho total de deslocamento para os eixos de percurso programados obtendo um número de trechos de mesmo comprimento (segmentação de curso eqüidistante). Internamente cada trecho corresponde à um bloco.

Número de cursos

Para a estampagem o primeiro curso é realizado no ponto final do primeiro trecho, já para o puncionamento é realizado no ponto de partida do primeiro trecho. Através do trecho total temos como resultado os seguintes números:

Estampagem: Número de cursos = número de trechos

Puncionamento: Número de cursos = número de trechos + 1

Funções auxiliares

As funções auxiliares são executadas no primeiro bloco gerado.

Sintaxe

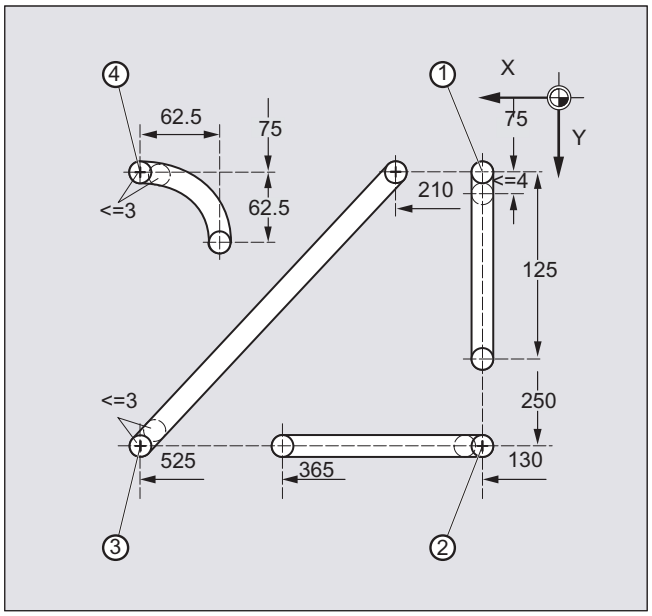
```
SPP=
SPN=
```

Significado

SPP	Tamanho do trecho (distância máxima de curso); ativado modalmente
SPN	Número de trechos por bloco; ativado por blocos

Exemplo 1

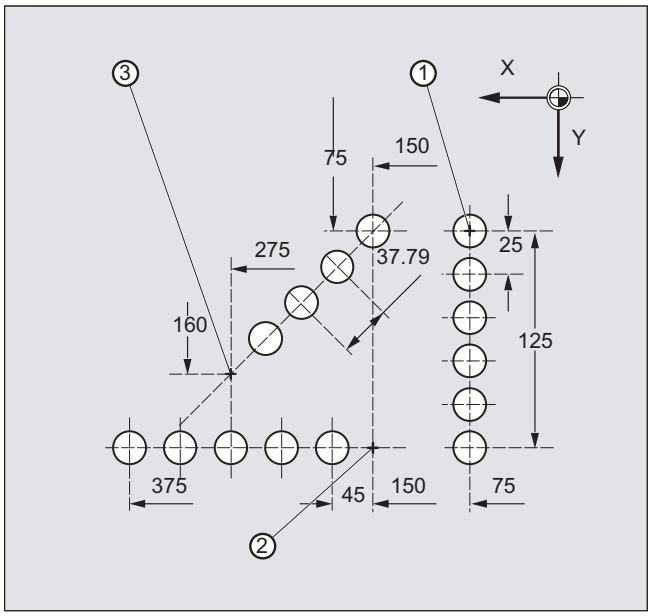
Os trechos de puncionamento programados devem ser automaticamente segmentados em trechos de mesmo tamanho.



Código de programa	**Comentário**
N100 G90 X130 Y75 F60 SPOF	; Posicionar no ponto de partida 1
N110 G91 Y125 SPP=4 SON	; Puncionamento ativado; comprimento máximo do trecho para segmentação automática do curso: 4 mm
N120 G90 Y250 SPOF	; Puncionamento desativado, posicionar no ponto de partida 2
N130 X365 SON	; Puncionamento ativado; comprimento máximo do trecho para segmentação automática do curso: 4 mm
N140 X525 SPOF	; Puncionamento desativado, posicionar no ponto de partida 3
N150 X210 Y75 SPP=3 SON	; Puncionamento ativado; comprimento máximo do trecho para segmentação automática do curso: 3 mm
N160 X525 SPOF	; Puncionamento desativado, posicionar no ponto de partida 4
N170 G02 X-62.5 Y62.5 I J62.5 SPP=3 SON	; Puncionamento ativado; comprimento máximo do trecho para segmentação automática do curso: 3 mm
N180 G00 G90 Y300 SPOF	; Puncionamento desativado

Exemplo 2

Para as fileiras individuais de furos deve ser realizada uma segmentação automática do curso. Para a segmentação é especificado o comprimento máximo do trecho (valor SPP).



Código de programa	**Comentário**
N100 G90 X75 Y75 F60 PON	; Posicionar no ponto de partida 1; estampagem ativada; puncionar furo individual
N110 G91 Y125 SPP=25	; Comprimento máximo do trecho para segmentação automática do curso: 25 mm
N120 G90 X150 SPOF	; Estampagem desativada, posicionar no ponto de partida 2
N130 X375 SPP=45 PON	; Estampagem ativada, comprimento máximo do trecho para segmentação automática do curso: 45 mm
N140 X275 Y160 SPOF	; Estampagem desativada, posicionar no ponto de partida 3
N150 X150 Y75 SPP=40 PON	; Estampagem ativada; no lugar do comprimento do trecho programado de 40 mm é utilizado o comprimento calculado de 37,79.
N160 G00 Y300 SPOF	; Estampagem desligada; posicionamento

12.2.1 Segmentação de curso para eixos de percurso

Comprimento do trecho SPP

Com o `SPP` especificamos a distância máxima do curso e com isso o comprimento máximo do trecho com que o trecho total de deslocamento deverá ser segmentado. A desativação do comando é feita com `SPOF` ou `SPP=0`.

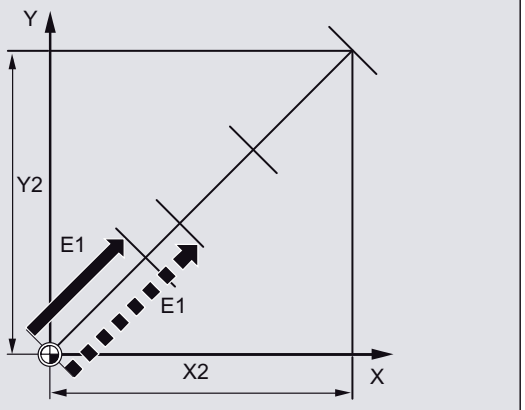
Exemplo:

```
N10 SON X0 Y0
N20 SPP=2 X10
```

O trecho total de deslocamento de 10 mm é segmentado em 5 trechos de 2 mm cada (`SPP=2`).

Indicação

A segmentação de curso com `SPP` sempre é realizada de forma eqüidistante: todos trechos têm o mesmo comprimento. Isto significa que o tamanho programado do trecho (valor do `SPP`) somente será aplicado se o quociente do trecho total de deslocamento e o valor SPP forem números inteiros. Se não for este o caso, então o tamanho do trecho será reduzido internamente de modo que obtenha um quociente de número inteiro.



X2/Y2	Percurso programado (bloco de puncionamento ou estampagem)
E1	Comprimento programado de trecho
E1'	Comprimento de trecho arredondado automaticamente

Exemplo:

```
N10 G1 G91 SON X10 Y10
N20 SPP=3.5 X15 Y15
```

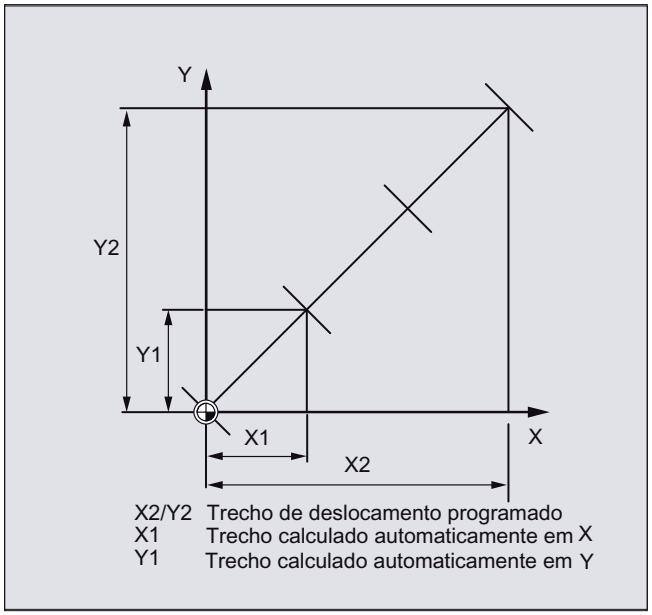
Com um trecho total de deslocamento de 15 mm e um trecho com comprimento de 3,5 mm não é obtido um quociente de número inteiro (4.28). Assim é realizada uma redução do valor SPP até o quociente de número inteiro mais próximo. Neste caso temos como resultado um trecho com comprimento de 3mm.

Número de trechos SPN

Com SPN definimos o número de trechos que devem ser gerados a partir do trecho total de deslocamento. O comprimento dos trechos é calculado automaticamente. Visto que o SPN é ativado por blocos, antes se deve ativar a estampagem ou o puncionamento com PON ou SON.

SPP e SPN no mesmo bloco

Se em um bloco programarmos tanto o comprimento do trecho (SPP) como o número de trechos (SPN), então para este bloco será aplicado o SPN, e para os demais o SPP. Se o SPP já foi ativado antes do SPN, então este torna-se ativo novamente após o bloco com SPN.



Indicação

Sempre que a estampagem/puncionamento estiver disponível no comando, então a programação da segmentação automática do curso com SPN ou SPP também não dependerá desta tecnologia.

12.2.2 Segmentação de curso para eixos individuais

Se além dos eixos de percurso também forem definidos eixos individuais como eixo de estampagem/puncionamento, então estes também podem estar submetidos à segmentação automática de curso.

Comportamento do eixo individual com SPP

O comprimento programado do trecho (`SPP`) sempre é relativo aos eixos de percurso. Por isso que o valor SPP é ignorado em um bloco onde não existe nenhum eixo de percurso programado além do movimento do eixo individual e do valor SPP.

Se forem programados tanto o eixo individual como o eixo de percurso, então o comportamento do eixo individual orienta-se conforme o ajuste do respectivo dado da máquina.

1. Ajuste padrão

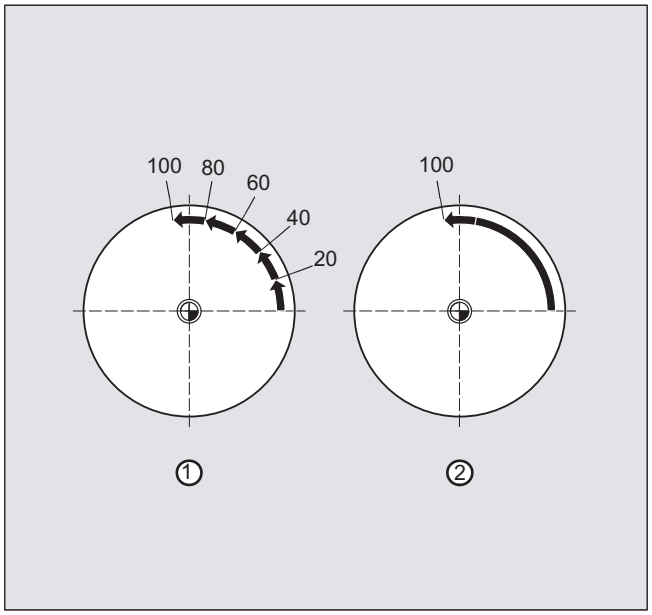
   O curso do eixo individual é segmentado uniformemente pelos blocos intermediários gerados pelo `SPP`.

Exemplo:

```
N10 G1 SON X10 A0
N20 SPP=3 X25 A100
```

Com um trecho de 3 mm são gerados 5 blocos a partir de um trecho total de deslocamento do eixo X (eixo de percurso) de 15 mm.

O eixo A faz um giro de 20° em cada bloco.



1. Eixo individual sem segmentação de curso

   O eixo individual movimenta-se pelo seu curso total no primeiro dos blocos gerados.

2. Segmentação de curso diferenciada

   O comportamento do eixo individual depende da interpolação dos eixos de percurso:

- Interpolação circular: Segmentação de curso
- Interpolação linear: sem segmentação de curso

Comportamento para SPN

O número programado de trechos também será aplicado se ao mesmo tempo não for programado um eixo de percurso.

Requisito: Eixo individual definido como eixo de estampagem/puncionamento.

Retificação 13

13.1 Monitoração de ferramenta específica de retificação no programa de peça (TMON, TMOF)

Função

Com o comando TMON podemos ativar a monitoração de geometria e rotação das ferramentas de retificar (tipo 400 - 499) no programa de peça NC. A monitoração permanece ativa, até que ela seja desativada no programa de peça através do comando TMOF.

Indicação

Observe as instruções do fabricante da máquina!

Requisitos

Os parâmetros de ferramenta $TC_TPG1 até $TC_TPG9 específicos de retificação precisam estar definidos.

Sintaxe

```
TMON(<n° T>)
TMOF(<n° T>)
```

Significado

TMON	Comando para **ativar** a monitoração específica de ferramentas de retificar
TMOF	Comando para **desativar** a monitoração específica de ferramentas de retificar
<n° T>	Indicação do número T **Nota:** Somente necessário se a ferramenta não estiver ativa com este número T.
TMOF(0)	Desativar a monitoração para todas ferramentas

Outras informações

Parâmetros específicos de ferramenta de retificar

Parâmetros	Significado	Tipo de dado
$TC_TPG1	Número do fuso	INT
$TC_TPG2	Diretriz de encadeamento Os parâmetros são mantidos idênticos para o lado esquerdo e direito do rebolo.	INT
$TC_TPG3	Raio mínimo do rebolo	REAL
$TC_TPG4	Largura mínima do rebolo	REAL
$TC_TPG5	Atual largura do rebolo	REAL
$TC_TPG6	Rotação máxima	REAL
$TC_TPG7	Velocidade periférica máxima	REAL
$TC_TPG8	Ângulo do rebolo inclinado	REAL
$TC_TPG9	Número de parâmetro para cálculo de raio	INT

Literatura:
Manual de funções básicas; Correção de ferramenta (W1)

Ativar a monitoração de ferramenta através da seleção de ferramenta

Em função de um dado de máquina se pode ativar de forma implícita a monitoração de ferramentas de retificar (tipo 400 - 499) através da seleção da ferramenta.

Em qualquer momento apenas **uma** monitoração pode estar ativada para cada fuso.

Monitoração de geometria

São monitorados o atual raio e a atual largura do rebolo.

A monitoração do valor nominal de rotação quanto ao limite de rotação é realizada de forma cíclica sob consideração do Override do fuso.

Como valor limite de rotação se aplica o menor valor que resulta da comparação entre a rotação máxima com a rotação calculada da velocidade periférica máxima do rebolo e do atual raio do rebolo.

Trabalhar sem números T e D

Por dado de máquina pode ser ajustado um número **T** padrão e um número **D** padrão,

que não precisa mais ser programado e que está ativo após o Power On / Reset.

Exemplo: Trabalhar com o mesmo rebolo

Através do dado de máquina se pode ajustar que a ferramenta ativa seja mantida mesmo com o Reset (veja " Atribuição livre de números D, número de corte (Página 440) ").

Outras funções 14

14.1 Funções de eixo (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL)

Função

Por exemplo, o AXNAME é utilizado na criação de ciclos de uso geral se o nome dos eixos não forem conhecidos.

O AX é utilizado para a programação indireta de eixos geométricos e eixos sincronizados. Aqui o identificador de eixo é armazenado em uma variável de tipo AXIS ou fornecida por um comando como o AXNAME ou SPI.

O SPI é utilizado quando forem utilizadas funções de eixo para um fuso, p. ex. fuso sincronizado.

O AXTOSPI é utilizado para converter um identificador de eixo em um índice de fuso (função inversa ao SPI).

O AXSTRING é utilizado para converter um identificador de eixo (tipo de dado AXIS) em uma string (função inversa ao AXNAME).

O ISAXIS é utilizado em todos ciclos válidos para assegurar que um determinado eixo geométrico esteja presente e com isso evitar com que uma chamada $P_AXNX não seja cancelada com falha.

O MODAXVAL é utilizado para determinar a posição Modulo em eixos rotativos Modulo.

Sintaxe

```
AXNAME("String")
AX[AXNAME("String")]
SPI(n)
```

AXTOSPI(A) ou AXTOSPI(B) ou AXTOSPI(C)

```
AXSTRING(SPI(n))
ISAXIS(<número do eixo geométrico>)
<posição Modulo>=MODAXVAL(<eixo>,<posição de eixo>)
```

Significado

AXNAME	Converte uma string de entrada no identificador de eixo; a string de entrada deve conter o nome de eixo válido.
AX	Identificador variável de eixo
SPI	Converte o número de fuso em identificador de eixo; o parâmetro de transferência deve conter um número de fuso válido.
n	Número do fuso
AXTOSPI	Converte um identificador de eixo em um índice de fuso de tipo Integer. O AXTOSPI corresponde à função inversa ao SPI.
X, Y, Z	Identificador de eixo de tipo AXIS como variável ou constante
AXSTRING	Se retorna a string com o número de fuso atribuído.
ISAXIS	Verifica se o eixo geométrico especificado está presente.
MODAXVAL	Determina em eixos rotativos Modulo a posição Modulo; esta corresponde ao resto Modulo relacionado à área Modulo parametrizada (no ajuste padrão compreende a faixa de 0 a 360 graus; através do MD30340 MODULO_RANGE_START e MD30330 $MA_MODULO_RANGE se pode alterar o início e o tamanho da área Modulo).

Indicação

Ampliações SPI

A função de eixo SPI(n) também pode ser aplicada para a leitura e gravação de componentes de Frame. Com isso podem ser gravados Frames com a sintaxe `$P_PFRAME[SPI(1),TR]=2.22`.

Com a programação adicional de posições de eixo através do endereço `AX[SPI(1)]=<posição de eixo>` nenhum eixo pode ser movimentado. O requisito para isso é que o fuso esteja em modo de posicionamento ou em modo de eixo.

Exemplos

Exemplo 1: AXNAME, AX, ISAXIS

```
Código de programa                                  Comentário
OVRA[AXNAME("eixo transversal")]=10                  ; Override para eixo transversal
AX[AXNAME("eixo transversal")]=50.2                  ; Posição do final para eixo
                                                      transversal
OVRA[SPI(1)]=70                                     ; Override para fuso 1
AX[SPI(1)]=180                                      ; Posição final para fuso 1
IF ISAXIS(1)==FALSE GOTOF CONTINUA                  ; Abscissa presente?
AX[$P_AXN1]=100                                     ; Percorrer abscissa
CONTINUA:
```

Exemplo 2: AXSTRING

Na programação com AXSTRING[SPI(n)] o índice de eixo do eixo atribuído ao fuso não é mais retornado como número de fuso, mas retornada a string `"Sn"`.

Código de programa	Comentário
`AXSTRING[SPI(2)]`	`; É retornada a string "S2".`

Exemplo 3: MODAXVAL

Deve ser determinada a posição Modulo do eixo rotativo Modulo A.

O valor de saída para o cálculo é a posição de eixo 372.55.

A faixa Modulo parametrizada é de 0 a 360 graus:

MD30340 MODULO_RANGE_START = 0

MD30330 $MA_MODULO_RANGE = 360

Código de programa	Comentário
`R10=MODAXVAL(A,372.55)`	`; Posição Modulo calculada R10 = 12.55.`

Exemplo 4: MODAXVAL

Se o identificador de eixo programado não se referir à um eixo rotativo Modulo, então o valor a ser convertido (<posição de eixo>) será retornado como sem ser alterado.

Código de programa	Comentário
`R11=MODAXVAL(X,372.55)`	`; X é eixo linear; R11 = 372.55.`

14.2 Eixos geométricos comutáveis (GEOAX)

Função

Com a função "Eixos geométricos comutáveis" pode-se alterar o grupo de eixos geométricos do programa de peça que foi configurado através de dados de máquina. Aqui um eixo de canal pode ser definido como eixo adicional e sincronizado como um eixo geométrico qualquer.

Sintaxe

```
GEOAX(<n>,<eixo de canal>,<n>,<eixo de canal>,<n>,<eixo de canal>)
GEOAX()
```

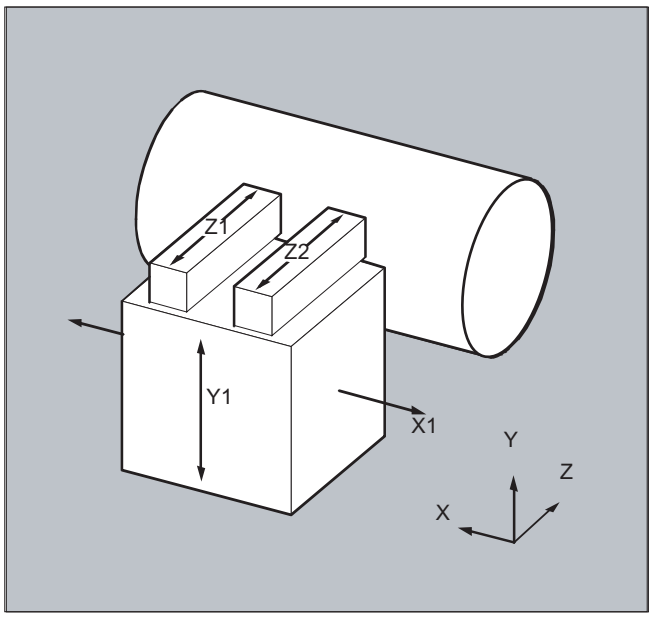
Significado

`GEOAX(...)`	Comando para comutação de eixos geométricos **Nota:** `GEOAX()` sem indicação de parâmetro chama a configuração básica dos eixos geométricos.
`<n>`	Com este parâmetro se indica o número do eixo geométrico que deve ser associado ao eixo de canal seguinte especificado. Faixa de valores: 1, 2 ou 3 **Nota:** Com <n>=0 se pode remover do grupo de eixos geométricos o eixo de canal seguinte indicado sem repor um substituto.
`<eixo de canal>`	Com este parâmetro se especifica o nome do eixo de canal que deve ser incorporado no grupo de eixos geométricos.

Exemplos

Exemplo 1: Ativar dois eixos alternadamente como eixo geométrico

Um carro de ferramentas pode ser movido através dos eixos de canal X1, Y1, Z1, Z2:



Os eixos geométricos são projetados de forma que, depois de ligar o sistema, seja ativado primeiro o Z1 como 3º eixo geométrico que recebe o nome de eixo geométrico "Z", formando um grupo de eixos geométricos com o X1 e o Y1.

Agora no programa de peça os eixos Z1 e Z2 devem ser aplicados alternadamente como eixo geométrico Z:

```
Código de programa          Comentário
...
N100 GEOAX(3,Z2)            ; Como 3° eixo geométrico (Z) atua o eixo de canal Z2.
N110 G1 ...
N120 GEOAX(3,Z1)            ; Como 3° eixo geométrico (Z) atua o eixo de canal Z1.
...
```

Exemplo 2: Comutação dos eixos geométricos com 6 eixos de canal

Uma máquina possui 6 eixos de canal com os nomes XX, YY, ZZ, U, V e W.

O ajuste básico da configuração de eixos geométricos através de dados de máquina é:

Eixo de canal XX = 1º eixo geométrico (eixo X)

Eixo de canal YY = 2º eixo geométrico (eixo Y)

Eixo de canal ZZ = 3º eixo geométrico (eixo Z)

Código de programa	**Comentário**
N10 GEOAX()	; A configuração básica dos eixos geométricos está ativa.
N20 G0 X0 Y0 Z0 U0 V0 W0	; Todos eixos em avanço rápido até a posição 0.
N30 GEOAX(1,U,2,V,3,W)	; O eixo de canal U passa para primeiro (X), V para segundo (Y) e W para terceiro eixo geométrico (Z).
N40 GEOAX(1,XX,3,ZZ)	; O eixo de canal XX passa para primeiro (X), ZZ para terceiro eixo geométrico (Z). O eixo de canal V permanece como segundo eixo geométrico (Y).
N50 G17 G2 X20 I10 F1000	; Círculo inteiro no plano X/Y. São movimentados os eixos de canal XX e V.
N60 GEOAX(2,W)	; O eixo de canal W passa a ser o segundo eixo geométrico (Y).
N80 G17 G2 X20 I10 F1000	; Círculo inteiro no plano X/Y. São movimentados os eixos de canal XX e W.
N90 GEOAX()	; Resetamento para o estado inicial.
N100 GEOAX(1,U,2,V,3,W)	; O eixo de canal U passa para primeiro (X), V para segundo (Y) e W para terceiro eixo geométrico (Z).
N110 G1 X10 Y10 Z10 XX=25	; Os eixos de canal U, V e W movimentam-se cada um até a posição 10. O XX como eixo adicional movimenta-se até a posição 25.
N120 GEOAX(0,V)	; O V é retirado do grupo de eixos geométricos. O U e W continuam sendo o primeiro (X) e o terceiro eixo geométrico (Z). O segundo eixo geométrico (Y) permanece desocupado.
N130 GEOAX(1,U,2,V,3,W)	; O eixo de canal U permanece como primeiro (X), o V passa a ser o segundo (Y) e o W passa a ser o terceiro eixo geométrico (Z).
N140 GEOAX(3,V)	; O V passa para o terceiro eixo geométrico (Z), onde o W é sobrescrito e com isso retirado do grupo de eixos geométricos. O segundo eixo geométrico (Y) ainda continua como antes, desocupado.

Indicação

Configuração de eixo

A associação entre os eixos geométricos, eixos adicionais, eixos de canal e eixos da máquina, assim como a definição dos nomes dos diversos tipos de eixo são realizados através dos seguintes dados de máquina:

MD20050 $MC_AXCONF_GEOAX_ASIGN_TAB (atribuição de eixo geométrico com

eixo de canal)

MD20060 $MC_AXCONF_GEOAX_NAME_TAB (nome de eixo geométrico no canal)

MD20070 $MC_AXCONF_MACHAX_USED (número de eixo de máquina válido no canal)

MD20080 $MC_AXCONF_CHANAX_NAME_TAB (nome de eixo de canal no canal)

MD10000 $MN_AXCONF_MACHAX_NAME_TAB (nome de eixo de máquina)

MD35000 $MA_SPIND_ASSIGN_TO_MACHAX (associação do fuso com o eixo de máquina)

Literatura:
Manual de funções básicas; Eixos, Sistemas de coordenadas, Frames (K2)

Restrições

- A comutação dos eixos geométricos é possível com:
  - transformação ativa
  - interpolação Spline ativa
  - correção ativa do raio da ferramenta
  - correção fina ativa da ferramenta
- Se eixos geométricos e eixos de canal tiverem o mesmo nome, não será possível realizar nenhuma troca do eixo geométrico correspondente.
- Nenhum dos eixos envolvidos com a comutação pode ser utilizado em uma ação que continuar além dos limites do bloco, como é possível, por exemplo, em eixos de posicionamento do tipo A ou em eixos escravos.
- Com o comando GEOAX somente podem ser substituídos os eixos geométricos presentes na ativação (portanto não se define nenhum novo).
- Uma troca de eixos com GEOAX durante a preparação da tabela de contorno (CONTPRON, CONTDCON) gera um alarme.

Condições gerais

Estado do eixo após a substituição

Um eixo substituído no grupo de eixos geométricos através da comutação pode ser programado como eixo adicional através de seu nome de eixo de canal e após o processo de comutação.

Frames, áreas de proteção, limites da área de trabalho

Com a comutação dos eixos geométricos são deletados todos os Frames, áreas de proteção e limites da área de trabalho.

Coordenadas polares

Uma troca de eixos geométricos com `GEOAX` ajusta as coordenadas polares modais para o valor 0 de forma similar à numa mudança de planos com `G17`-`G19`.

DRF, NPV

Um eventual deslocamento com manivela eletrônica (DRF) ou um deslocamento externo do ponto zero permanece ativo após a comutação.

Configuração básica dos eixos geométricos

O comando `GEOAX()` chama a configuração básica do grupo de eixos geométricos.

Após o POWER ON e na mudança para o modo de operação "Aproximação do ponto de referência" se restabelece automaticamente a configuração básica.

Correção do comprimento da ferramenta

Uma correção ativa do comprimento da ferramenta também está ativa após o processo de comutação. Porém, ela vale como ainda não executada para os eixos geométricos que foram adotados ou trocados de posição. No primeiro comando de movimento para estes eixos geométricos o percurso resultante é composto da soma da correção do comprimento da ferramenta e o percurso programado.

Os eixos geométricos que mantém sua posição no grupo de eixos em uma comutação, também mantém seu estado relativo à correção do comprimento da ferramenta.

Configuração de eixos geométricos com transformação ativa

A configuração de eixos geométricos válida (definida através de dados de máquina) em uma transformação ativa não pode ser alterada através da função "Eixos geométricos comutáveis".

Se houver a necessidade de alterar a configuração dos eixos geométricos em relação às transformações, isso somente será possível com outra transformação.

Uma configuração de eixos geométricos alterada através do `GEOAX` é cancelada com a ativação de uma transformação.

Se os ajustes dos dados da máquina forem contraditórios para a transformação e para a comutação de eixos geométricos, então os ajustes da transformação terão prioridade.

Exemplo:

Isto se uma transformação estiver ativa. De acordo com os dados de máquina a transformação em um RESET deve ser mantida, mas ao mesmo tempo deve ser estabelecida a configuração básica dos eixos geométricos com um RESET. Neste caso a configuração de eixos geométricos será mantida como foi definida com a transformação.

14.3 Contentor de eixo (AXCTSWE, AXCTSWED)

Função

Em máquinas Transfer rotativas/máquinas de cabeçotes múltiplos os eixos portadores de peças de trabalho movem-se de uma unidade usinagem para outra. Como as unidades de usinagem estão submetidas à diversos canais NCU, no caso de uma mudança de estação/posição dos eixos portadores de peças, se deve realizar uma nova atribuição dinâmica com o canal NCU correspondente. Para este objetivo é que servem os contentores de eixos.

Em um determinado momento sempre se encontra ativo apenas um eixo de fixação/fuso da peça de trabalho na unidade de usinagem local. O contentor de eixos agrupa as opções de ligação para todos os eixos de fixação/fusos, dos quais sempre se encontra ativo apenas e exatamente um para a unidade de usinagem.

A mudança dos eixos utilizáveis e definidos através de um contentor de eixos é realizada através o deslocamento dos registros no contentor de eixos ("Rotação de contentor de eixo") através de um passo especificado (número de Slots) através de dado de máquina.

A chamada da rotação do contentor de eixo a partir do programa de peça é realizada com o comando `AXCTSWE` ou `AXCTSWED`.

Sintaxe

```
AXCTSWE(<contentor de eixo>)
AXCTSWED(<contentor de eixo>)
```

Significado

`AXCTSWE`	Comando para rotação de um contentor de eixo Quando no comando as liberações de todos canais dos eixos do contentor estiverem congeladas, será realizada a rotação do contentor com o incremento armazenado no SD41700 $SN_AXCT_SWWIDTH[<número do contentor>].
`AXCTSWED`	Comando pada rotação de um contentor de eixo sob efeito isolado do canal ativo (variante de comando para a colocação em funcionamento!) **Nota:** Os eixos registrados no contentor somente são liberados se nos demais canais que possuírem eixos no contentor, estiverem em estado de Reset.

`<contentor de eixo>`	Identificador do contentor de eixo que deve ser transferido. As possíveis indicações são:	
	`CT<número do contentor>`	Na combinação de letras CT se anexa o número do contentor de eixo. Exemplo: `CT3`
	`<nome do contentor>`	Nome do contentor de eixo ajustado individualmente com MD12750 $MN_AXCT_NAME_TAB. Exemplo: `A_CONT3`

Outras informações

Contentor de eixo

Através de contentores de eixos podem ser realizadas as seguintes atribuições:

- eixos locais e/ou
- Eixos lincados

Os contentores de eixos com eixos lincados (de ligação) são um meio de operação usado além da NCU (NCU global), que é coordenado pelo comando. São possíveis os contentores de eixos onde são administrados exclusivamente eixos locais.

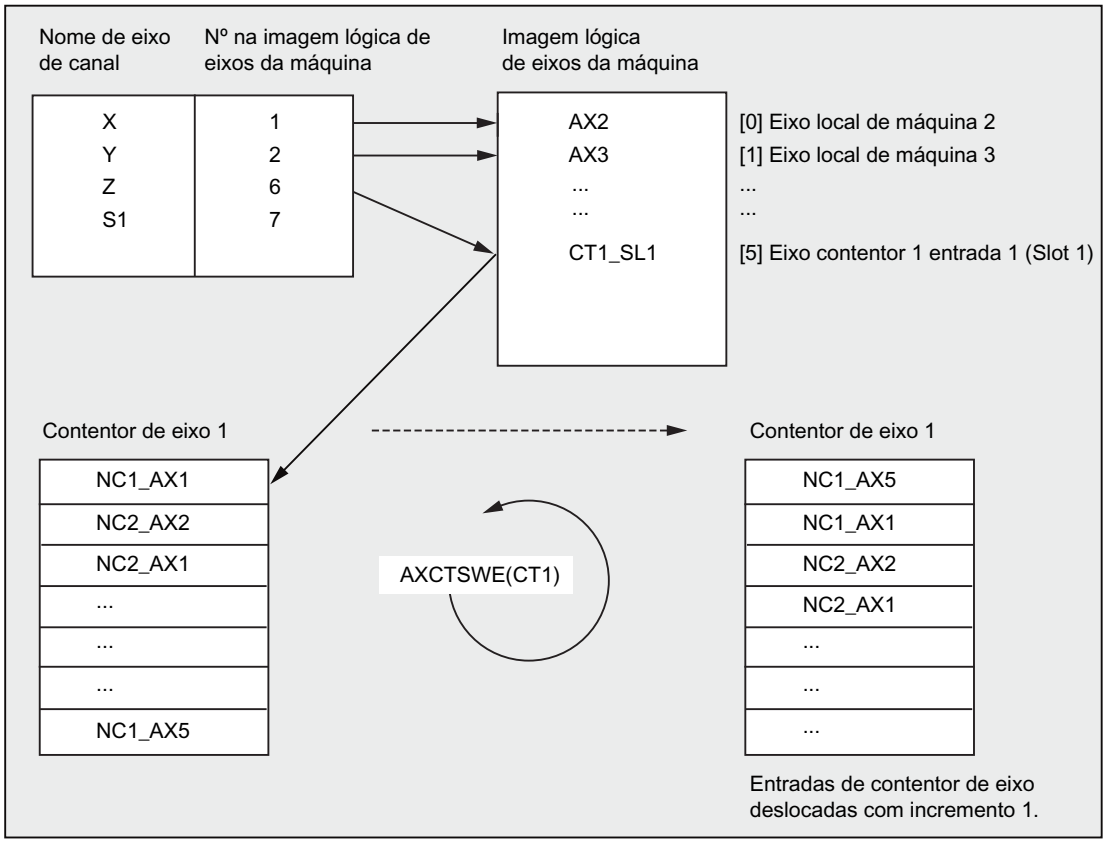
Literatura:
Para instruções detalhadas sobre a projeção de contentores de eixos, veja:
Manual de funções ampliadas; Vários painéis de operação em várias NCUs, Sistemas descentralizados (B3)

Critérios de liberação

AXCTSWE( )

Todo canal, cujos eixos estão registrados no contentor especificado, permite a liberação para uma rotação de contentor (enable) assim que ele concluir a execução da posição/estação. Quando no comando as liberações de **todos** canais dos eixos do contentor estiverem congeladas, será realizada a rotação do contentor com o incremento armazenado no SD41700 $SN_AXCT_SWWIDTH[<número do contentor>].

Exemplo:



Após a rotação de contentor de eixo 1, no eixo de canal Z, é realizada a atribuição do eixo AX5 à NCU1 ao invés do AX1 à NCU1.

AXCTSWED( )

A variante de comando `AXCTSWED()` pode ser aplicada para simplificar a colocação em funcionamento. O contentor de eixos gira apenas sob efeito isolado do canal ativo com o incremento específico de contentor armazenado na SD41700 $SN_AXCT_SWWIDTH[<número de contentor>]. Esta chamada somente deve ser utilizada quando os demais canais com eixos no contentor estiverem em estado **Reset**.

Efeito

A nova atribuição de eixos após uma rotação de contentor de eixos tem efeito sobre todas NCUs cujos canais estão relacionados com o contentor de eixos girado, através da tela lógica de eixos de máquina.

Rotação de contentor de eixo com GET/GETD implícito

Na liberação de uma rotação de contentor de eixo são atribuídos todos os eixos de contentor do canal através do GET ou GETD. Uma entrega dos eixos somente será permitida após a rotação do contentor de eixo.

Indicação

Este comportamento pode ser ajustado através de dado de máquina. Observe as instruções do fabricante da máquina.

Indicação

A rotação de contentor de eixo cp, GET / GETD implícito **não** pode ser aplicado para um eixo em estado de eixo de processamento principal (p. ex. para um eixo PLC), pois este eixo deve se desfazer deste estado no caso da rotação de contentor de eixo.

14.4 Espera pela posição de eixo válida (WAITENC)

Função

Com o comando de linguagem WAITENC é possível esperar no programa NC até que os eixos configurados com o dado MD34800 $MA_WAIT_ENC_VALID = 1 estejam nas posições de eixo síncronas e restauradas.

Em estado de espera pode ocorrer uma interrupção, p. ex. através da partida de uma ASUP ou pela mudança de modos de operação para JOG. Com a continuação do programa o estado de espera é novamente cancelado.

Indicação

O estado de espera é indicado na interface de operação através do estado de parada "Esperar pelo sistema de medição".

Sintaxe

O WAITENC pode ser programado na seção de programa de qualquer programa NC.

A programação deve ser realizada em um bloco próprio

```
...
WAITENC
...
```

Exemplo

Por exemplo, o WAITENC é utilizado no programa de usuário controlado por eventos .../_N_CMA_DIR/_N_PROG_EVENT_SPF, como mostra o seguinte exemplo de aplicação.

Exemplo de aplicação:Retrocesso de ferramenta após POWER OFF com transformação de orientação

Uma usinagem com orientação de ferramenta foi interrompida através de queda de energia. Em seguida, ao inicializar o sistema é ativado o programa de usuário controlado por eventos .../_N_CMA_DIR/_N_PROG_EVENT_SPF.

No programa de usuário controlado por eventos existe uma espera pela sincronização e restauração das posições de eixo através do WAITENC, e depois para processar um Frame, que alinha o WCS no sentido da ferramenta.

Código de programa	Comentário
...	
IF $P_PROG_EVENT == 4	; Inicialização.
IF $P_TRAFO <> 0	; A transformação foi ativada.
WAITENC	; Espera pelas posições de eixo válida dos eixos de orientação.
TOROTZ	; Rotação do eixo Z do WCS no sentido do eixo de ferramenta.
ENDIF	
M17	
ENDIF	
...	

Em seguida, em modo de operação JOG, a ferramenta pode ser afastada através de um movimento de retrocesso no sentido do eixo da ferramenta.

14.5 Controlar o escopo disponível de linguagem NC (STRINGIS)

Função

Com a função `STRINGIS(...)` é possível verificar se a String especificada como elemento da linguagem de programação NC está disponível no atual escopo de linguagem.

Definição

```
INT STRINGIS(STRING <nome>)
```

Sintaxe

```
STRINGIS(<nome>)
```

Significado

`STRINGIS`:	Função com valor de retorno
`<nome>`:	Nome do elemento verificado da linguagem de programação NC
Valor de retorno:	O formato do valor de retorno é yxx (decimal).

Elementos da linguagem de programação NC

Os seguintes elementos da linguagem de programação NC podem ser verificados:

- Códigos G de todos os grupos de função G existentes, p. ex. `G0`, `INVCW`, `POLY`, `ROT`, `KONT`, `SOFT`, `CUT2D`, `CDON`, `RMB`, `SPATH`
- Endereços DIN ou NC como por exemplo `ADIS`, `RNDM`, `SPN`, `SR`, `MEAS`
- Funções, p. ex. `TANG(...)` ou `GETMDACT`
- Procedimentos, p. ex. `SBLOF`.
- Palavras-chave, p. ex. `ACN`, `DEFINE` ou `SETMS`
- Dados de sistema, p. ex. dados de máquina `$M`... , dados de ajuste `$S`... ou dados de opção `$O`...
- Variável de sistema `$A`... , `$V`... , `$P`...
- Parâmetros de cálculo `R`...
- Nomes de ciclo de ciclos ativados
- Variáveis GUD e LUD
- Nomes de macro
- Nomes de Label

Valor de retorno

Um valor de retorno somente é relevante nas primeiras 3 casas decimais. O formato do valor de retorno é yxx, com y = informação básica e xx = informação de detalhe.

Valor de retorno	Significado	
000	A String 'name' não é reconhecida pelo presente sistema [1)]	
100	A String 'name' é um elemento da linguagem de programação NC, mas atualmente não é programável (opção/função inativa)	
2xx	A String 'name' é um elemento programável da linguagem de programação NC (opção/função ativa). A informação de detalhe xx contém outras informações sobre o tipo do elemento:	
	xx	**Significado**
	01	Endereço DIN ou endereço NC [2)]
	02	Código G (p. ex. `G04`, `INVCW`)
	03	Função com valor de retorno
	04	Função sem valor de retorno
	05	Palavra-chave (p. ex. `DEFINE`)
	06	Dados de máquina (`$M`...), dados de ajuste (`$S`...) ou dados de opção `$O`...)
	07	Parâmetros de sistema, p. ex. variável de sistema `$`...) ou parâmetro de cálculo `R`...)
	08	Ciclo (O ciclo deve estar carregado no NCK e os programas de ciclo ativados [3)])
	09	Variável GUD (A variável GUD deve estar definida nos arquivos de definição GUD e ativada nas variáveis GUD)
	10	Nome de macro (A macro estar definida nos arquivos de definição de macro; as macros devem estar ativadas) [4)]
	11	Variável LUD do atual programa de peça
	12	Código G ISO (O modo de linguagem ISO deve estar ativo)
400	A String 'name' é um endereço NC, que não foi reconhecido como xx == 01 ou xx == 10 e que não é G nem R [2)]	
y00	Nenhuma atribuição específica possível	

1) Dependendo do comando, sob determinadas circunstâncias, apenas uma menor quantidade de comandos de linguagem NC da Siemens são reconhecidos, SINUMERIK 802D sl. Nestes comandos numéricos é retornado o valor 0 para as Strings que são, em princípio, comandos de linguagem NC da Siemens. Este comportamento pode ser alterado através do MD10711 $MN_NC_LANGUAGE_CONFIGURATION. Com MD10711 = 1 sempre é retornado o valor 100 para comandos de linguagem NC da Siemens.

2) Os endereços NC são as seguintes letras: A, B, C, E, I, J, K, Q, U, V, W, X, Y, Z. Estes endereços NC também podem ser programados com uma extensão de endereço. A extensão de endereço pode ser especificada na verificação realizada com STRINGIS. Exemplo: `201 == STRINGIS("A1")`.
As letras: D, F, H, L, M, N, O, P, S, T são endereços NC ou funções auxiliares que são usadas por definição de usuário. Para elas sempre é retornado o valor 400. Exemplo: 400 == STRINGIS("D"). Estes endereços NC não podem ser especificados com extensão de endereço na verificação realizada com STRINGIS.
Exemplo: `000 == STRINGIS("M02")`, mas `400 == STRINGIS("M")`.

3) Os nomes de parâmetros de ciclos não podem ser verificados com STRINGIS.
4) Endereço definido como macro, por exemplo o G, H, M e L são identificados como macro

Exemplos

Nos seguintes exemplos considera-se que o elemento de linguagem NC especificado como String seja programável no comando numérico, isto se não houver outra observação.

1. A String "T" é definida como função auxiliar:
   ```
   400 == STRINGIS("T")
   000 == STRINGIS ("T3")
   ```
2. A String "X" é definida como eixo:
   ```
   201 == STRINGIS("X")
   201 == STRINGIS("X1")
   ```
3. A String "A2" é definida como endereço NC com extensão:
   ```
   201 == STRINGIS("A")
   201 == STRINGIS("A2")
   ```
4. A String "INVCW" é definida como código G nomeado:
   ```
   202 == STRINGIS("INVCW")
   ```
5. A String "$MC_GCODE_RESET_VALUES" é definida como dado de máquina:
   ```
   206 == STRINGIS("$MC_GCODE_RESET_VALUES")
   ```
6. A String "GETMDACT" é uma função de linguagem NC:
   ```
   203 == STRINGIS("GETMDACT ")
   ```
7. A String "DEFINE" é uma palavra-chave:
   ```
   205 == STRINGIS("DEFINE")
   ```
8. A String "$TC_DP3" é um parâmetro de sistema (componente do comprimento da ferramenta):
   ```
   207 == STRINGIS("$TC_DP3")
   ```
9. A String "$TC_TP4" é um parâmetro de sistema (tamanho da ferramenta):
   ```
   207 == STRINGIS("$TC_TP4")
   ```
10. A String "$TC_MPP4" é um parâmetro de sistema (estado de alojamento do magazine):
    - O gerenciamento do magazine de ferramentas está ativo. `207 == STRINGIS("$TC_MPP4")` ;
    - O gerenciamento do magazine de ferramentas não está ativo. `000 == STRINGIS("$TC_MPP4")`

    Veja também o parágrafo abaixo: Gerenciamento de magazine de ferramentas.
11. A String "MACHINERY_NAME" é definida como variável GUD:
    ```
    209 == STRINGIS("MACHINERY_NAME")
    ```
12. A String "LONGMACRO" é definida como macro:
    ```
    210 == STRINGIS("LONGMACRO")
    ```
13. A String "MYVAR" é definida como variável LUD:
    ```
    211 == STRINGIS("MYVAR")
    ```
14. A String "XYZ" não é nenhum comando, variável GUS, nome de macro ou de ciclo conhecido no NCK:
    ```
    000 == STRINGIS("XYZ")
    ```

Gerenciamento de magazine de ferramentas

Se a função de gerenciamento de magazine de ferramentas não estiver ativa, o STRINGIS retorna para os parâmetros de sistema do gerenciamento do magazine de ferramentas, independentemente do dado de máquina:

- MD10711 $MN_NC_LANGUAGE_CONFIGURATION

sempre o valor 000.

Modo ISO

Se a função "Modo ISO" estiver ativa:

- MD18800 $MN_MM_EXTERN_LANGUAGE (ativação de linguagens NC externas)
- MD10880 $MN_ MM_EXTERN_CNC_SYSTEM (para sistema de comando que deve ser adaptado)

o STRINGIS verifica a String indicada primeiro como código G do SINUMERIK. Se a String não for nenhum código G do SINUMERIK, então ela será verificada como código G de linguagem ISO.

As comutações programadas (`G290` (modo SINUMERIK), `G291` (modo ISO)) não têm nenhum efeito sobre o STRINGIS.

Exemplo

Os dados de máquina relevantes para o STRINGIS(...) possuem os seguintes valores:

- MD10711 $MN_NC_LANGUAGE_CONFIGURATION = 2 (Somente são reconhecidos os comandos de linguagem NC cujas opções estiverem definidas)
- MD19410 $ON_TRAFO_TYPE_MASK = 'H0' (Opção: Transformações)
- MD10700 $MN_PREPROCESSING_LEVEL='H43' (Pré-processamento de ciclos ativo)

O próximo programa de exemplo é processado sem mensagem de erro:

```
Código de programa                      Comentário
N1 R1=STRINGIS("TRACYL")                ;   R1 == 0, pois o TRACYL, por causa da falta de
                                        ;   opção de transformação, é tido como "não reconhecido"
                                        ;   no reconhecimento
N2 IF STRINGIS("TRACYL") == 204         ;
N3 TRACYL(1,2,3)                        ;   N3 é saltado
N4 ELSE
N5 G00                                  ;   e ao invés disso é executado o N5
N6 ENDIF
N7 M30
```

14.6 Chamada de função ISVAR e leitura do índice Array de dados de máquina

Função

O comando ISVAR é uma função para linguagem NC com um:

- Valor de função do tipo BOOL
- Parâmetro de transferência do tipo STRING

O comando ISVAR retorna TRUE se o parâmetro de transferência contém uma variável conhecida pelo NC (dado de máquina, dado de ajuste, variável de sistema, variáveis gerais como GUD's).

Sintaxe

```
ISVAR(<identificador de variável>)
ISVAR(<identificador>,[<valor>,<valor>])
```

Significado

`<identificador de variável>`	Os parâmetros de transferência do tipo string podem ser não dimensionais, monodimensionais ou bidimensionais.
`<identificador>`	Identificador com uma variável com ou sem índice Array conhecida pelo NC como dado de máquina, dado de ajuste, variável de sistema ou variável geral. **Ampliação:** Para dados de máquina gerais e específicos de canal é lido o primeiro elemento do Array mesmo com o índice faltando
`<valor>`	Valor de função do tipo BOOL

Verificações

De acordo com o parâmetro de transferência são realizadas as seguintes verificações:

- O identificador está presente
- Trata-se de um campo monodimensional ou bidimensional
- É permitido um índice Array

O TRUE somente é retornado se todas estas verificações resultarem de forma positiva. Se apenas uma das verificações não for preenchida ou ocorrer um erro de sintaxe, então esta será confirmada com FALSE. Variáveis axiais são aceitas como índice para os nomes de eixo, mas não serão verificadas posteriormente.

Ampliação: Leitura de Array sem índice de dados de máquina e de dados de ajuste.

Na falta do índice dos dados de máquina **gerais e específicos de canal** o alarme 12400 "Canal % 1 Bloco % 2 Campo % 3 Elemento não disponível" **não será mais** emitido.

Ainda assim se deve programar **pelo menos o índice de eixo** com dados de máquina **específicos de eixo**. Do contrário se emite o alarme 12400.

Exemplo: Chamada de função ISVAR

```
Código de programa                                  Comentário
DEF INT VAR1
DEF BOOL IS_VAR=FALSE                               ; O parâmetro de transferência é uma variável geral
N10 IS_VAR=ISVAR("VAR1")                            ; Neste caso o IS_VAR é TRUE
DEF REAL VARARRAY[10,10]
DEF BOOL IS_VAR=FALSE                               ; diversas variantes de sintaxe
N20 IS_VAR=ISVAR("VARARRAY[,]")                     ; O IS_VAR é TRUE com um
                                                      Array bidimensional
N30 IS_VAR=ISVAR("VARARRAY")                        ; O IS_VAR é TRUE, a variável existe
N40 IS_VAR=ISVAR("VARARRAY[8,11]")                  ; O IS_VAR é FALSE, o índice de Array não é permitido
N50 IS_VAR=ISVAR("VARARRAY[8,8")                    ; O IS_VAR é FALSE, erro de sintaxe para "]"
                                                      que falta
N60 IS_VAR=ISVAR("VARARRAY[,8]")                    ; O IS_VAR é TRUE, o índice de Array é permitido
N70 IS_VAR=ISVAR("VARARRAY[8,]")                    ; O IS_VAR é TRUE
DEF BOOL IS_VAR=FALSE                               ; O parâmetro de transferência é um dado de máquina
N100 IS_VAR=ISVAR("$MC_GCODE_RESET_VALUES[1]"       ; O IS_VAR é TRUE
DEF BOOL IS_VAR=FALSE                               ; O parâmetro de transferência é uma variável de
                                                      sistema
N10 IS_VAR=ISVAR("$P_EP")                           ; Neste caso o IS_VAR é TRUE
N10 IS_VAR=ISVAR("$P_EP[X]")                        ; Neste caso o IS_VAR é TRUE
```

Exemplo: Leitura de Array de dados de máquina com ou sem índice

O primeiro elemento é lido com

```
R1=$MC_EXTERN_GCODE_RESET_VALUES
```

isto corresponde como anteriormente

```
R1=$MC_EXTERN_GCODE_RESET_VALUES[0]
```

ou se lê o primeiro elemento

```
R1=$MA_POSTCTRL_GAIN[X1]
```

isto corresponde como anteriormente

```
R1=$MA_POSTCTRL_GAIN[0, X1]
```

Também se lê o primeiro elemento em ações sincronizadas com

```
WHEN TRUE DO $R1 = $MC_EXTERN_GCODE_RESET_VALUES
```

isto corresponde como anteriormente

```
WHEN TRUE DO $R1 = $MC_EXTERN_GCODE_RESET_VALUES[0]
```

e, como anteriormente, **não é lido** e vem acompanhado do alarme 12400.

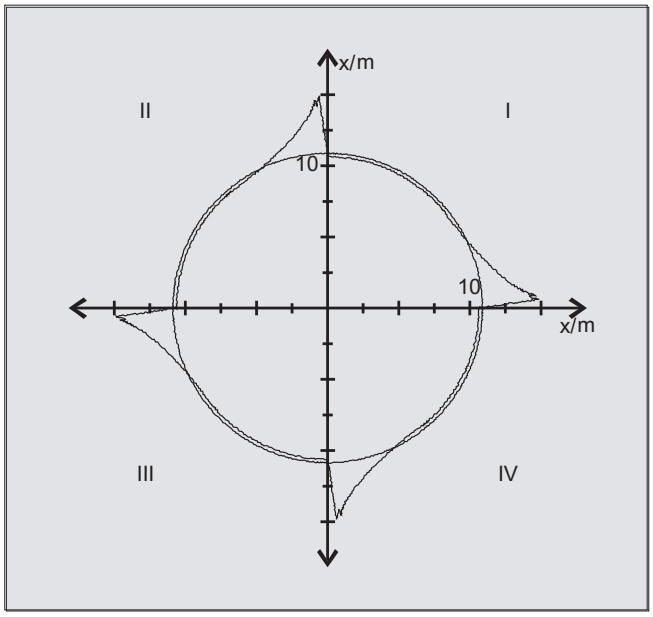
O alarme 12400 continua sendo emitido com

```
R1=$MA_POSTCTRL_GAIN
```

14.7 Adaptação de curvas características de compensação (QECLRNON, QECLRNOF)

Função com erro! Marca de texto não definida.

A compensação de erro de quadrante (QFK) reduz os erros de contorno que são gerados na reversão do sentido de deslocamento através de irregularidade mecânica (p. ex. atrito, folga) ou torção. Com base em uma rede neural do comando, os dados ideais de compensação são aprendidos durante um fase de aprendizado e com isso as curvas características de compensação são obtidas automaticamente. O aprendizado pode ser realizado simultaneamente com até 4 eixos.



Sintaxe

```
QECLRNON
QECLRNOF
```

Ativação do processo de aprendizado: QECLRNON

O processo de aprendizado propriamente dito é ativado no programa NC através do comando `QECLRNON` e com a indicação dos eixos:

```
QECLRNON (X1, Y1, Z1, Q)
```

Somente quando este comando estiver ativo é que as curvas características serão alteradas.

Desativação do aprendizado: QECLRNOF

Depois que os movimentos de aprendizado dos eixos solicitados forem concluídos, o processo de aprendizado será desativado com `QECLRNOF` para todos eixos simultaneamente.

Significado

`QECLRNON (eixo.1,…4)`	Ativação da função "Aprendizado da compensação de erro de quadrante"
`QECLRNO`	Desativação da função "Aprendizado da compensação de erro de quadrante"
`QECLRN.SPF`	Ciclo de aprendizado
`QECDAT.MPF`	Programa NC modelo para atribuição das variáveis de sistema e para a parametrização do ciclo de aprendizado.
`QECTEST.MPF`	Programa NC modelo para teste de forma circular

Descrição

Os movimentos de deslocamento necessários para a adaptação são gerados com o auxílio do programa NC. Neste estão os movimentos de adaptação na forma de ciclos de adaptação.

Primeira adaptação

Para a primeira adaptação durante a colocação em funcionamento existem programas NC modelos armazenados no disquete do programa básico do PLC, estes servem para a adaptação de movimentos assim como para a atribuição das variáveis de sistema QFK:

Readaptação

Uma nova otimização das curvas características já adaptadas é possível com a "readaptação". Neste caso são usados os dados que se encontram na memória do usuário. Para a readaptação adapte os programas NC modelos conforme necessidade.

Os parâmetros do ciclo de adaptação (p. ex. QECLRN.SPF) também devem ser modificados para a "adaptação":

- Definir "Modo de adaptação" = 1
- Se necessário, reduzir o "número de adaptações"
- Se necessário, ativar a "adaptação modular" e definir os limites de área correspondentes

14.8 Chamada interativa a partir do programa de peça (MMC)

Função

Através do comando MMC podem ser exibidas, a partir do programa de peça, janelas de diálogo (telas de diálogo) definidas pelo usuário na HMI.

A aparência das janelas de diálogo é definida através de uma configuração puramente textual (arquivo COM no diretório de ciclos), o software de sistema HMI permanece inalterado.

As janelas de diálogo definidas pelo usuário não podem ser chamadas simultaneamente em diversos canais.

Sintaxe

```
MMC(CYCLES,PICTURE_ON,T_SK.COM,BILD,MGUD.DEF,BILD_3.AWB,TEST_1,A1","
S")
```

Significado

MMC	A partir do programa de peça chamar a janela de diálogo interativa na HMI.
CYCLES	Área de operação na qual são executados os diálogos de usuário configurados.
PICTURE_ON ou PICTURE_OFF	Comando: Seleção e desseleção de telas
T_SK.COM	Arquivo Com: Nome do arquivo da tela de diálogo (ciclos do usuário). Aqui se define a aparência das telas de diálogo. Na tela de diálogo podem ser exibidas variáveis de sistema e/ou textos de comentário.
TELA	Nome da tela de diálogo: As diversas telas são selecionadas mediante nomes de telas de diálogo.
MGUD.DEF	Arquivo de definição de dados do usuário acessado durante a gravação/leitura de variáveis.
TELA_3.AWB	Arquivo de figura
TEST_1	Tempo de exibição ou variável de confirmação
A1	Variáveis de texto...",
"S"	Modo de confirmação: sincronizado, confirmação através da softkey "OK"

Literatura

Para informações detalhadas sobre a programação do comando MMC (incl. com exemplos de programação), veja o manual de colocação em funcionamento.

14.9 Tempo de processamento do programa / contador de peças

14.9.1 Tempo de processamento do programa / contador de peças (resumo)

Para suporte ao operador da máquina-ferramenta são disponibilizadas informações sobre o tempo de processamento e a quantidade de peças produzidas.

Estas informações podem ser processadas como variáveis de sistema no programa NC e/ou PLC. Simultaneamente estas informações são disponibilizadas para exibição na interface de operação.

14.9.2 Tempo de processamento do programa

Função

A função "tempo de processamento do programa" disponibiliza um relógio interno do NC para supervisão de processos tecnológicos, que pode ser lido através de variáveis de sistema de NC e específicos de canal em programas de peça e em ações sincronizadas.

O disparo para medição do tempo de processamento ($AC_PROG_NET_TIME_TRIGGER) é a única variável de sistema gravável da função e serve para medição seletiva de segmentos de programa. Isto significa que através da gravação do disparo no programa NC se pode ativar e desativar a medição do tempo.

Variável de sistema	Significado	Atividade
Específico de NC		
$AN_SETUP_TIME	Tempo desde a última inicialização do comando com valores Standard ("partida a frio") em minutos É automaticamente resetado para "0" com valores Standard em cada inicialização do comando.	• sempre ativo
$AN_POWERON_TIME	Tempo desde a última inicialização normal do comando ("reinicialização") em minutos É automaticamente resetado para "0" em cada inicialização normal do comando.	

Variável de sistema	Significado	Atividade
Específico de canal		
$AC_OPERATING_TIME	Tempo total de processamento de programas NC em modo automático, dado em segundos O valor é automaticamente resetado para "0" em cada inicialização do comando.	• Ativação através do MD27860 • apenas modo de operação AUTOMÁTICO
$AC_CYCLE_TIME	Tempo de processamento do programa NC selecionado, em segundos O valor é automaticamente resetado para "0" com a partida de um novo programa NC.	
$AC_CUTTING_TIME	Tempo de processamento em segundos Aqui se mede o tempo de movimento dos eixos de percurso (pelo menos um está ativo) sem avanço rápido ativo em todos programas NC entre a partida do NC e o fim do programa / NC-Reset. A medição também é interrompida se o tempo de espera estiver ativo. O valor é automaticamente resetado para "0" com valores Standard em cada inicialização do comando.	
$AC_ACT_PROG_NET_TIME	Atual tempo de processamento líquido do atual programa NC em segundos É automaticamente resetado para "0" com a partida de um programa NC.	• sempre ativo • apenas modo de operação AUTOMÁTICO
$AC_OLD_PROG_NET_TIME	Tempo líquido de processamento do programa encerrado corretamente com `M30` indicado em segundos	
$AC_OLD_PROG_NET_TIME_COUNT	Alterações no $AC_OLD_PROG_NET_TIME Após o POWER ON o $AC_OLD_PROG_NET_TIME_COUNT está em "0". $AC_OLD_PROG_NET_TIME_COUNT sempre é aumentado, se o comando gravar novamente o $AC_OLD_PROG_NET_TIME.	

Variável de sistema	Significado		Atividade
$AC_PROG_NET_TIME_TRIGGER	Disparo para medição do tempo de processamento:		• apenas modo de operação AUTOMÁTICO
	0	Estado neutro O disparo não está ativo.	
	1	Finalizar Encerra a medição e copia o valor do $AC_ACT_PROG_NET_TIME para o $AC_OLD_PROG_NET_TIME. O $AC_ACT_PROG_NET_TIME é passado para "0" e continua sendo processado depois disso.	
	2	Iniciar Inicia a medição e com isso passa o $AC_ACT_PROG_NET_TIME para "0". O $AC_OLD_PROG_NET_TIME não é alterado.	
	3	Stop Cessa a medição. Não altera o $AC_OLD_PROG_NET_TIME e mantém o $AC_ACT_PROG_NET_TIME sempre constante até a continuação.	
	4	Continuação Continuação da medição, isto é, se retoma uma medição parada anteriormente. O $AC_ACT_PROG_NET_TIME continua a ser processado. O $AC_OLD_PROG_NET_TIME não é alterado.	
Através do POWER ON todas variáveis de sistema são passadas para "0"!			
Literatura: Uma descrição detalhada das variáveis de sistema listadas está disponível no(a): Manual de funções básicas; BAG, Canal, Operação do programa, Comportamento Reset (K1), Capítulo: Tempo de processamento do programa			

Indicação

Fabricante da máquina

A ativação do temporizador ativável é realizada através do dado de máquina MD27860 $MC_PROCESSTIMER_MODE.

O comportamento das medições de tempo ativas em determinadas funções (p. ex. GOTOS, Override = 0%, avanço de teste ativo, teste de programa, ASUP, PROG_EVENT, …) é configurado através dos dados de máquina MD27850 $MC_PROG_NET_TIMER_MODE e MD27860 $MC_PROCESSTIMER_MODE.

Literatura:
Manual de funções básicas; BAG, Canal, Operação do programa, Comportamento Reset (K1), Capítulo: Tempo de processamento do programa

Indicação

Tempo restante para uma peça de trabalho

Quando peças de trabalho iguais são produzidas continuamente, pode-se com os valores de Timer:

- Tempo de processamento da última peça de trabalho produzida (veja $AC_OLD_PROG_NET_TIME)

e

- Atual tempo de processamento (veja $AC_ACT_PROG_NET_TIME)

determinar o tempo restante para uma peça de trabalho.

O tempo restante é indicado na interface de operação, junto com o atual tempo de processamento.

ATENÇÃO

Utilização do STOPRE

As variáveis de sistema $AC_OLD_PROG_NET_TIME e $AC_OLD_PROG_NET_TIME_COUNT não geram nenhuma parada implícita de pré-processamento. Na utilização no programa de peça isto não é crítico, se o valor das variáveis de sistema são herdadas da execução de programa anterior. Entretanto, se o disparo para medição de tempo de processamento ($AC_PROG_NET_TIME_TRIGGER) for gravado com alta frequência e com isso o $AC_OLD_PROG_NET_TIME for alterado com muita frequência, então se deve utilizar um `STOPRE` implícito no programa de peça.

Condições gerais

- **Localização de blocos**

  Na localização de blocos não é determinado nenhum tempo de processamento do programa.

- **REPOS**

  O tempo de um processo REPOS é considerado no cálculo do atual tempo de processamento ($AC_ACT_PROG_NET_TIME).

Exemplos

Exemplo 1: Medição do tempo do "mySubProgrammA"

Código de programa

```
...
N50 DO $AC_PROG_NET_TIME_TRIGGER=2
N60 FOR ii= 0 TO 300
N70 mySubProgrammA
N80 DO $AC_PROG_NET_TIME_TRIGGER=1
N95 ENDFOR
N97 mySubProgrammB
N98 M30
```

Depois que o programa processar a linha `N80`, no $AC_OLD_PROG_NET_TIME estará gravado o tempo líquido de processamento do "mySubProgrammA".

O valor do $AC_OLD_PROG_NET_TIME:

- é mantido além do `M30`.
- é atualizado depois de cada execução de loop.

Exemplo 2: Medição de tempo do "mySubProgrammA" e do "mySubProgrammC"

Código de programa

```
...
N10 DO $AC_PROG_NET_TIME_TRIGGER=2
N20 mySubProgrammA
N30 DO $AC_PROG_NET_TIME_TRIGGER=3
N40 mySubProgrammB
N50 DO $AC_PROG_NET_TIME_TRIGGER=4
N60 mySubProgrammC
N70 DO $AC_PROG_NET_TIME_TRIGGER=1
N80 mySubProgrammD
N90 M30
```

14.9.3 Contador de peças

Função

A função "Contador de peças" disponibiliza diversos contadores, que podem ser utilizados especialmente para a contagem de peças, interna do comando.

Os contadores existem como variáveis de sistema específicas de canal com acesso de gravação e leitura, em uma faixa de valores de 0 até 999 999 999.

Variável de sistema	Significado
$AC_REQUIRED_PARTS	Número de peças que devem ser produzidas (número de peças nominal) Neste contador pode-se definir o número de peças que, ao ser alcançado, passa o número de peças real ($AC_ACTUAL_PARTS) para "0"..
$AC_TOTAL_PARTS	Número total de peças produzidas (número de peças real total) Este contador indica o número de todas peças produzidas desde o momento da partida. O valor é automaticamente resetado para "0" com valores Standard somente com uma inicialização do comando.
$AC_ACTUAL_PARTS	Número de peças produzidas (número de peças real) Neste contador é registrado o número de todas peças produzidas desde o momento da partida. Ao ser alcançado o número de peças nominal ($AC_REQUIRED_PARTS) o contador é automaticamente passado para "0" ($AC_REQUIRED_PARTS > 0 é a condição prévia).
$AC_SPECIAL_PARTS	Número de peças contado pelo usuário Este contador permite ao usuário uma contagem de peças conforme sua própria definição. Uma emissão de alarme pode ser definida com o alcance do número de peças nominal ($AC_REQUIRED_PARTS). O zeramento do contador deve ser realizado pelo próprio usuário.

Indicação

Todos contadores de peças são passados para "0" com valores padrão na fase de inicialização do comando e podem ser lidos e gravados independentemente de sua ativação.

Indicação

Através de dados de máquina específicos de canal pode-se influenciar sobre a ativação de contadores, o momento do zeramento e o algoritmo de contagem.

Indicação

Contagem de peças com comando M definido por usuário

Os dados de máquina podem ser ajustados de modo que os impulsos de contagem para os diversos contadores de peças sejam disparados através de comandos M definidos pelo usuário ao invés do fim do programa `M2`/`M30`.

Literatura

Mais informações sobre a função "Contador de peças" estão disponíveis no(a):

- Manual de funções básicas; BAG, Canal, Operação do programa, Comportamento Reset (K1), Capítulo: Contador de peças

14.10 Alarmes (SETAL)

Função

Em um programa NC podem ser introduzidos alarmes. Estes são representados na interface de operação por um campo especial. Com um alarme sempre está associada uma reação do comando em função da categoria do alarme.

Literatura:
Para mais informações sobre reações de alarmes, veja o manual de colocação em funcionamento.

Sintaxe

```
SETAL(<número do alarme>)
SETAL(<número do alarme>,<seqüência de caracteres>)
```

Significado

`SETAL`	Palavra-chave para programação de um alarme. O `SETAL` deve ser programado em um bloco NC separado.
`<número de alarme>`	Variável de tipo INT. Contém o número do alarme. A faixa válida de números de alarme está entre 60000 e 69999, sendo que os números do 60000 ao 64999 estão reservados para ciclos da SIEMENS e do 65000 ao 69999 estão disponíveis para o usuário.
`<seqüência de caracteres>`	Na programação de alarmes de ciclos de usuário também se pode indicar uma seqüência de caracteres de até 4 parâmetros. Nestes parâmetros podem ser definidos textos variáveis de usuário. Entretanto, também estão disponíveis os seguintes parâmetros pré-definidos:

Parâmetros	Significado
%1	Número de canal
%2	Número do bloco, Label
%3	Índice de texto para alarmes de ciclo
%4	parâmetros de alarme adicionais

Indicação

Os textos de alarme devem ser configurados na interface de operação.

Exemplo

Código de programa	Comentário
...	
N100 SETAL(65000)	; Define o alarme nº 65000
...	

Programas de desbaste próprios 15

15.1 Funções de apoio para o desbaste

Funções

Para a operação de desbaste nos são oferecidos ciclos de usinagem prontos para serem aplicados. Além disso existe a possibilidade de se utilizar as seguintes funções para criação de próprios programas de desbaste:

- Criar tabela de contorno (CONTPRON)
- Criar tabela de contorno codificada (CONTDCON)
- Desativar a preparação de contorno (EXECUTE)
- Determinar a intersecção entre dois elementos de contorno (INTERSEC).

  (Apenas para tabelas que foram criadas com CONTPRON.)
- Executar por blocos os elementos de contorno de uma tabela (EXECTAB)

  (Apenas para tabelas que foram criadas com CONTPRON.)
- Calcular dados de círculos (CALCDAT)

Indicação

Estas funções podem ser utilizadas não apenas para o desbaste, mas de modo universal.

Requisitos

Antes da chamada das funções CONTPRON ou CONTDCON se deve:

- aproximar um ponto de partida que permita uma usinagem livre de colisões.
- desativar a correção do raio da ferramenta com `G40`.

15.2 Criar tabela de contorno (CONTPRON)

Função

Com o comando `CONTPRON` é ativada a preparação de contorno. Os blocos NC chamados em seguida não serão executados, mas divididos em movimentos individuais e armazenados na tabela de contorno. Cada um dos elementos de contorno corresponde à uma linha na tabela em um campo bidimensional da tabela de contorno. É retornado o número de detalonados determinados.

Sintaxe

Ativar a preparação de contorno:
```
CONTPRON(<tabela do contorno>,<tipo de usinagem>,<detalonados>,
<sentido de usinagem>)
```
Desativar a preparação de contorno e retornar ao modo de execução normal:
```
EXECUTE(<ERRO>)
```
Veja " Desativar a preparação de contorno (EXECUTE) "

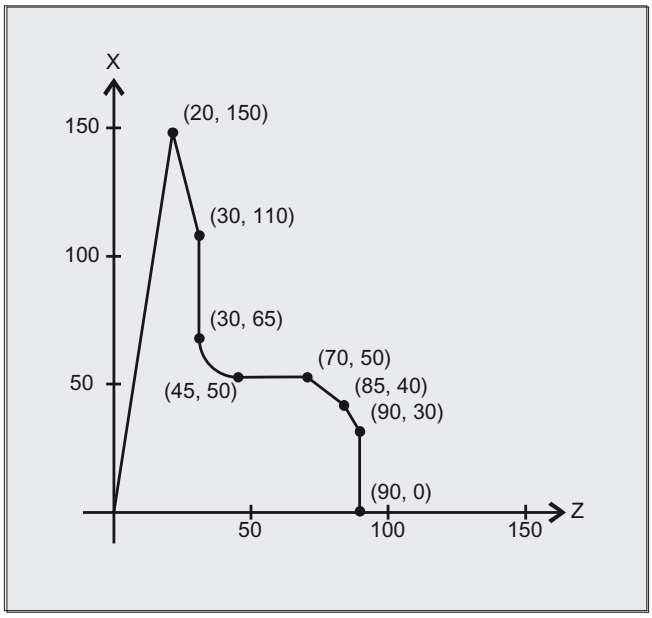
Significado

`CONTPRON`	Comando para ativar a preparação de contorno para criação de uma tabela de contorno		
`<tabela de contorno>`	Nome da tabela de contorno		
`<tipo de usinagem>`	Parâmetro para o tipo de usinagem		
	Tipo:	CHAR	
	Valor:	`"G"`	Torneamento longitudinal: Usinagem interna
		`"L"`	Torneamento longitudinal: Usinagem externa
		`"N"`	Torneamento transversal: Usinagem interna
		`"P"`	Torneamento transversal: Usinagem externa
`<detalonados>`	Variável de resultado para o número de elementos de detalonado ocorridos		
	Tipo:	INT	
`<sentido de usinagem>`	Parâmetro para o sentido da usinagem		
	Tipo:	INT	
	Valor:	0	Preparação de contorno para frente (valor padrão)
		1	Preparação de contorno nos dois sentidos

Exemplo 1

Criação de uma tabela de contorno com:

- nome "KTAB"
- máx. 30 elementos de contorno (círculos, retas)
- uma variável para o número de elementos de detalonado ocorridos
- uma variável para mensagens de erro



Programa NC:

Código de programa	Comentário
N10 DEF REAL KTAB[30,11]	; Tabela de contorno de nome KATB e com o máx. de 30 elementos de contorno, o valor de parâmetro 11 (número de colunas da tabela) é um tamanho fixo.
N20 DEF INT NUMDETALONADOS	; Variável para a quantidade de elementos de detalonado de nome NUMDETALONADOS.
N30 DEF INT ERRO	; Variável para resposta de erro (0=nenhum erro, 1=erro).
N40 G18	
N50 CONTPRON(KTAB,"G",NUMDETALONADOS)	; Ativar a preparação de contorno.
N60 G1 X150 Z20	; N60 até N120: Descrição do contorno
N70 X110 Z30	
N80 X50 RND=15	
N90 Z70	
N100 X40 Z85	
N110 X30 Z90	
N120 X0	

Código de programa	**Comentário**
`N130 EXECUTE(ERRO)`	`; Finalizar o preenchimento da tabela do contorno, mudança para o modo normal do programa.`
`N140 …`	`; Continuação do processamento da tabela.`

Tabela de contorno KTAB:

Índice Linha	Coluna									
(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
7	7	11	0	0	20	150	0	82.40535663	0	0
0	2	11	20	150	30	110	-1111	104.0362435	0	0
1	3	11	30	110	30	65	0	90	0	0
2	4	13	30	65	45	50	0	180	45	65
3	5	11	45	50	70	50	0	0	0	0
4	6	11	70	50	85	40	0	146.3099325	0	0
5	7	11	85	40	90	30	0	116.5650512	0	0
6	0	11	90	30	90	0	0	90	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

Explicação do conteúdo das colunas:

(0) Ponteiro no elemento de contorno seguinte (no número de linha do próprio)

(1) Ponteiro no elemento de contorno anterior

(2) Codificação do modo de contorno para o movimento

Possíveis valores para X = abc

a = 10^2 G90 = 0 G91 = 1

b = 10^1 G70 = 0 G71 = 1

c = 10^0 G0 = 0 G1 = 1 G2 = 2 G3 = 3

(3), (4) Ponto inicial dos elementos de contorno

(3) = abscissa, (4) = ordenada no plano atual

(5), (6) Ponto final dos elementos de contorno

(5) = abscissa, (6) = ordenada no plano atual

(7) Indicador máx./mín.: identifica os máximos e mínimos no contorno

(8) Valor máximo entre o elemento de contorno e a abscissa (na usinagem longitudinal) ou ordenada (na usinagem transversal). O ângulo está em função do tipo de usinagem programado.

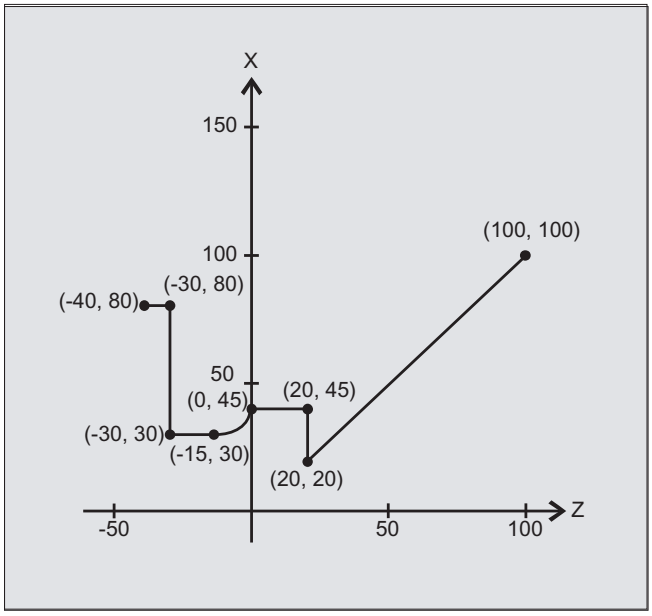
(9), (10) Coordenadas do centro do elemento de contorno, no caso de um bloco de circunferência.

(9) = abscissa, (10) = ordenada

Exemplo 2

Criação de uma tabela de contorno com

- nome KTAB
- máx. 92 elementos de contorno (círculos, retas)
- Modo de operação: Torneamento longitudinal, usinagem externa
- Preparação para frente e para trás



Programa NC:

Código de programa	Comentário
N10 DEF REAL KTAB[92,11]	; Tabela de contorno com nome KTAB e por exemplo máx. de 92 elementos de contorno; o valor de parâmetro 11 é um tamanho fixo.
N20 DEF CHAR BT="L"	; Tipo de operação para CONTPRON: Torneamento longitudinal, usinagem externa
N30 DEF INT HE=0	; Número de elementos de detalonado=0
N40 DEF INT MODE=1	; Preparação para frente e para trás
N50 DEF INT ERR=0	; Resposta de erro
...	
N100 G18 X100 Z100 F1000	
N105 CONTPRON(KTAB,BT,HE,MODE)	; Ativar a preparação de contorno.
N110 G1 G90 Z20 X20	
N120 X45	
N130 Z0	
N140 G2 Z-15 X30 K=AC(-15) I=AC(45)	
N150 G1 Z-30	

```
Código de programa                                  Comentário
N160 X80
N170 Z-40
N180 EXECUTE(ERR)                                   ; Finalizar o preenchimento da
                                                      tabela do contorno, mudança para o
                                                      modo normal do programa.
...
```

Tabela de contorno KTAB:

Após a finalização da preparação do contorno, o contorno se encontra disponível nos dois sentidos.

Índice	Coluna										
Linha	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
0	6[1)]	7[2)]	11	100	100	20	20	0	45	0	0
1	0[3)]	2	11	20	20	20	45	-3	90	0	0
2	1	3	11	20	45	0	45	0	0	0	0
3	2	4	12	0	45	-15	30	5	90	-15	45
4	3	5	11	-15	30	-30	30	0	0	0	0
5	4	7	11	-30	30	-30	45	-1111	90	0	0
6	7	0[4)]	11	-30	80	-40	80	0	0	0	0
7	5	6	11	-30	45	-30	80	0	90	0	0
8	1[5)]	2[6)]	0	0	0	0	0	0	0	0	0
	...										
83	84	0[7)]	11	20	45	20	80	0	90	0	0
84	90	83	11	20	20	20	45	-1111	90	0	0
85	0[8)]	86	11	-40	80	-30	80	0	0	0	0
86	85	87	11	-30	80	-30	30	88	90	0	0
87	86	88	11	-30	30	-15	30	0	0	0	0
88	87	89	13	-15	30	0	45	-90	90	-15	45
89	88	90	11	0	45	20	45	0	0	0	0
90	89	84	11	20	45	20	20	84	90	0	0
91	83[9)]	85[10)]	11	20	20	100	100	0	45	0	0

Explanação dos conteúdos de colunas e das observações sobre as linhas 0, 1, 6, 8, 83, 85 e 91

São aplicadas as explanações de conteúdo de coluna mencionadas no exemplo 1.

Sempre na linha de tabela 0:

1) Precedente: A linha n contém o fim do contorno para frente

2) Posterior: A linha n é o fim da tabela de contorno para frente

Em cada caso uma vez dentro dos elementos de contorno para frente:

3) Precedente: Início do contorno (para frente)

4) Posterior: Fim do contorno (para frente)

Sempre na linha do fim da tabela do contorno (para frente) +1:

5) Precedente: Número de detalonados para frente

6) Posterior: Número de detalonados para trás

Em cada caso uma vez dentro dos elementos de contorno para trás:

7) Posterior: Fim de contorno (para trás)

8) Precedente: Início de contorno (para trás)

Sempre na última linha da tabela:

9) Precedente: A linha n é o início da tabela de contorno (para trás)

10) Posterior: A linha n contém o início do contorno (para trás)

Outras informações

Comandos de deslocamento permitidos, sistema de coordenadas

Para programação do contorno são permitidos os seguintes comandos G:

- Grupo G 1: `G0`, `G1`, `G2`, `G3`

Também são possíveis:

- Arredondamento e chanfro
- Programação de círculo através do `CIP` e `CT`

As funções Spline, polinômio e rosca conduzem à erros.

Não são permitidas alterações do sistema de coordenadas através da ativação de um Frame entre `CONTPRON` e `EXECUTE`. O mesmo vale para uma mudança entre `G70` e `G71` ou `G700` e `G710`.

Uma troca dos eixos geométricos com `GEOAX` durante a preparação da tabela do contorno gera um alarme.

Elementos de detalonado

Opcionalmente a descrição do contorno dos elementos de detalonado pode ser realizada em uma subrotina ou em blocos individuais.

Desbaste em função do sentido de contorno programado

A preparação de contorno CONTPRON é ampliada de modo que, após sua chamada, a tabela do contorno esteja disponível independentemente do sentido programado.

15.3 Criar tabela de contorno codificada (CONTDCON)

Função

Na preparação de contorno ativado com CONTDCON os blocos NC seguintes são armazenados em uma tabela de contorno de 6 colunas que permite a economia de espaço. Cada elemento de contorno corresponde à uma linha na tabela do contorno. Conhecendo-se as regras de codificação especificadas, por exemplo, podemos compor ciclos através de programas de código DIN a partir das linhas da tabela. Na linha de tabela com o número 0 são armazenados os dados do ponto de saída.

Sintaxe

Ativar a preparação de contorno:
`CONTDCON(<tabela de contorno>,<sentido de usinagem>)`

Desativar a preparação de contorno e retornar ao modo de execução normal:
`EXECUTE(<ERRO>)`

Veja " Desativar a preparação de contorno (EXECUTE) "

Significado

`CONTDCON`	Comando para ativar a preparação de contorno para criação de uma tabela de contorno codificada	
`<tabela de contorno>`	Nome da tabela de contorno	
`<sentido de usinagem>`	Parâmetro para sentido de usinagem	
	Tipo:	INT
	Valor:	0 Preparação de contorno conforme seqüência de contornos (valor padrão)
		1 inadmissível

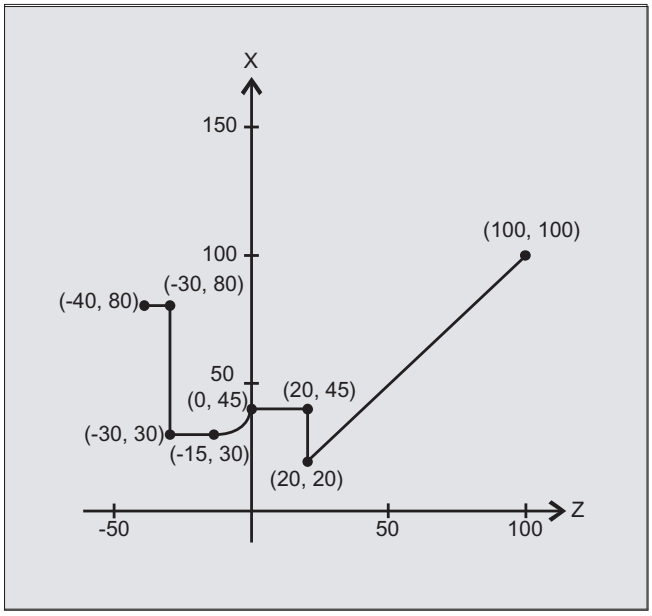
Indicação

Os códigos G permitidos para o CONTDCON no segmento de programa a ser tabelado são mais completos que no CONTPRON. Além disso, os avanços e o tipo de avanço podem ser salvos juntos por segmento de contorno.

Exemplo

Criação de uma tabela de contorno com:

- nome "KTAB"
- Elementos de contorno (círculos, retas)
- Modo de operação: Torneamento
- Sentido de usinagem: para frente



Programa NC:

Código de programa	**Comentário**
N10 DEF REAL KTAB[9,6]	; Tabela de contorno com nome KTAB e 9 linhas de tabela. Estas permitem 8 blocos de contorno. O valor de parâmetro 6 (número de colunas da tabela) é um tamanho fixo.
N20 DEF INT MODE = 0	; Variável para o sentido de usinagem. Valor padrão 0: somente no sentido programado do contorno.
N30 DEF INT ERROR = 0	; Variável para a resposta de erro.
...	
N100 G18 G64 G90 G94 G710	
N101 G1 Z100 X100 F1000	
N105 CONTDCON (KTAB, MODE)	; Chamada da preparação de contorno (o MODE não pode ser suprimido).
N110 G1 Z20 X20 F200	; Descrição do contorno.

```
Código de programa                                    Comentário
N120 G9 X45 F300
N130 Z0 F400
N140 G2 Z-15 X30 K=AC(-15) I=AC(45)F100
N150 G64 Z-30 F600
N160 X80 F700
N170 Z-40 F800
N180 EXECUTE(ERROR)                                   ; Finalizar o preenchimento da
                                                        tabela do contorno, mudança para
                                                        o modo normal do programa.
...
```

Tabela de contorno KTAB:

	Índice de coluna					
	0	1	2	3	4	5
Índice de linha	Modo de contorno	Ponto final Abscissa	Ponto final Ordenada	Centro Abscissa	Centro Ordenada	Avanço
0	30	100	100	0	0	7
1	11031	20	20	0	0	200
2	111031	20	45	0	0	300
3	11031	0	45	0	0	400
4	11032	-15	30	-15	45	100
5	11031	-30	30	0	0	600
6	11031	-30	80	0	0	700
7	11031	-40	80	0	0	800
8	0	0	0	0	0	0

Explicação do conteúdo das colunas:

Linha 0: Codificações para o **ponto de partida:**

Coluna 0:	10^0 (posição da unidade): G0 = 0
	10^1 (posição da dezena): G70 = 0, G71 = 1, G700 = 2, G710 = 3
Coluna 1:	Ponto de partida da abscissa
Coluna 2:	Ponto de partida da ordenada
Coluna 3-4:	0
Coluna 5:	Índice de linha do último segmento de contorno na tabela

Linhas 1-n: Registros dos **segmentos de contorno**

Coluna 0:	10^0 (posição da unidade): G0 = 0, G1 = 1, G2 = 2, G3 = 3
	10^1 (posição da dezena): G70 = 0, G71 = 1, G700 = 2, G710 = 3
	10^2 (posição da centena): G90 = 0, G91 = 1
	10^3 (posição da milhar): G93 = 0, G94 = 1, G95 = 2, G96 = 3
	10^4 (posição da dezena de milhar): G60 = 0, G44 = 1, G641 = 2, G642 = 3
	10^5 (posição da centena de milhar): G9 = 1
Coluna 1:	Ponto final da abscissa
Coluna 2:	Ponto final da ordenada
Coluna 3:	Centro da abscissa na interpolação circular
Coluna 4:	Centro da ordenada na interpolação circular
Coluna 5:	Avanço

Outras informações

Comandos de deslocamento permitidos, sistema de coordenadas

Para programação do contorno são permitidos os seguintes grupos G e comandos G:

Grupo G 1:	G0, G1, G2, G3
Grupo G 10:	G60, G64, G641, G642
Grupo G 11:	G9
Grupo G 13:	G70, G71, G700, G710
Grupo G 14:	G90, G91
Grupo G 15:	G93, G94, G95, G96, G961

Também são possíveis:

- Arredondamento e chanfro
- Programação de círculo através do CIP e CT

As funções Spline, polinômio e rosca conduzem à erros.

Não são permitidas alterações do sistema de coordenadas através da ativação de um Frame entre CONTDCON e EXECUTE. O mesmo vale para uma mudança entre G70 e G71 ou G700 e G710.

Uma troca dos eixos geométricos com GEOAX durante a preparação da tabela do contorno gera um alarme.

Sentido de usinagem

A tabela de contorno gerada com CONTDCON está prevista para o desbaste no sentido programado do contorno.

15.4 Determinar a intersecção entre dois elementos de contorno (INTERSEC).

Função

O INTERSEC determina a intersecção de dois elementos de contorno normalizados, obtidos da tabela de contorno gerada com CONTPRON.

Sintaxe

```
<estado>=INTERSEC(<tabela de contorno_1>[<elemento de contorno_1>],
<tabela de contorno_2>[<elemento de contorno_2>],<intersecção>,<tipo
de usinagem>)
```

Significado

INTERSEC	Palavra-chave para determinação da intersecção de dois elementos de contorno a partir de tabelas geradas com CONTPRON		
<estado>	Variável para o estado da intersecção		
	Tipo:	BOOL	
	Valor:	TRUE	intersecção encontrada
		FALSE	nenhuma intersecção encontrada
<tabela de contorno_1>	Nome da primeira tabela de contorno		
<elemento de contorno_1>	Número do elemento de contorno da primeira tabela de contorno		
<tabela de contorno_2>	Nome da segunda tabela de contorno		
<elemento de contorno_2>	Número do elemento de contorno da segunda tabela de contorno		
<intersecção>	Coordenadas da intersecção no plano ativo (G17 / G18 / G19)		
	Tipo:	REAL	
<tipo de usinagem>	Parâmetro para o tipo de usinagem		
	Tipo:	INT	
	Valor:	0	Cálculo da intersecção no plano ativo com o parâmetro 2 (valor padrão)
		1	Cálculo de intersecção independente do plano transferido

Indicação

Observe que as variáveis devem ser definidas antes de sua utilização.

A transferência dos contornos requer a conservação dos valores definidos com CONTPRON:

Parâmetros	Significado
2	Codificação do modo de contorno para o movimento
3	Ponto de início do contorno, abscissa
4	Ponto de início do contorno, ordenada
5	Ponto final do contorno, abscissa
6	Ponto final do contorno, ordenada
9	Coordenada do centro para a abscissa (somente para contorno circular)
10	Coordenada do centro para a ordenada (somente para contorno circular)

Exemplo

Determinar a intersecção do elemento de contorno 3 da tabela TABNAME1 com o elemento de contorno 7 da tabela TABNAME2. As coordenadas da intersecção no plano ativo são armazenadas na variável ISCOORD (1º elemento = abscissa, 2º elemento = ordenada). Se não existir nenhuma intersecção, ocorre um salto para NOINTERSECT (nenhuma intersecção encontrada).

```
Código de programa                                                  Comentário
DEF REAL TABNAME1[12,11]                                            ; Tabela de contorno 1
DEF REAL TABNAME2[10,11]                                            ; Tabela de contorno 2
DEF REAL ISCOORD[2]                                                 ; Variável para coordenadas da
                                                                      intersecção.
DEF BOOL ISPOINT                                                    ; Variável para estado da
                                                                      intersecção.
DEF INT MODE                                                        ; Variável para tipo de usinagem.
…
MODE=1                                                              ; Cálculo independente do plano
                                                                      ativo.
N10 ISPOINT=INTERSEC(TABNAME1[3],TABNAME2[7],ISCOORD,MODE)          ; Chamada da intersecção dos
                                                                      elementos de contorno.
N20 IF ISPOINT==FALSE GOTOF NOINTERSECT                             ; Salto até o NOINTERSECT.
…
```

15.5 Executar por blocos os elementos de contorno de uma tabela (EXECTAB)

Função

Com o comando EXECTAB podemos executar por blocos os elementos de contorno de uma tabela que, por exemplo, foi gerada com o comando CONTPRON.

Sintaxe

```
EXECTAB(<tabela de contorno>[<elemento de contorno>])
```

Significado

EXECTAB	Comando para execução de um elemento de contorno
<tabela de contorno>	Nome da tabela de contorno
<elemento de contorno>	Número do elemento de contorno

Exemplo

Os elementos de contorno 0 até 2 da tabela KTAB devem ser executados por blocos.

Código de programa	Comentário
N10 EXECTAB(KTAB[0])	; Executar o elemento 0 da tabela KTAB.
N20 EXECTAB(KTAB[1])	; Executar o elemento 1 da tabela KTAB.
N30 EXECTAB(KTAB[2])	; Executar o elemento 2 da tabela KTAB.

15.6 Calcular dados de círculos (CALCDAT)

Função

Como o comando `CALCDAT` podemos calcular o raio e as coordenadas do centro do círculo através de três ou quatro pontos conhecidos do círculo. Os pontos indicados devem ser diferentes um do outro. No caso de 4 pontos, que não estão exatamente posicionados sobre o círculo, é utilizado o valor médio para o cálculo do centro e o raio do círculo.

Sintaxe

```
<estado>=CALCDAT(<pontos do
círculo>[<número>,<tipo>],<número>,<resultado>)
```

Significado

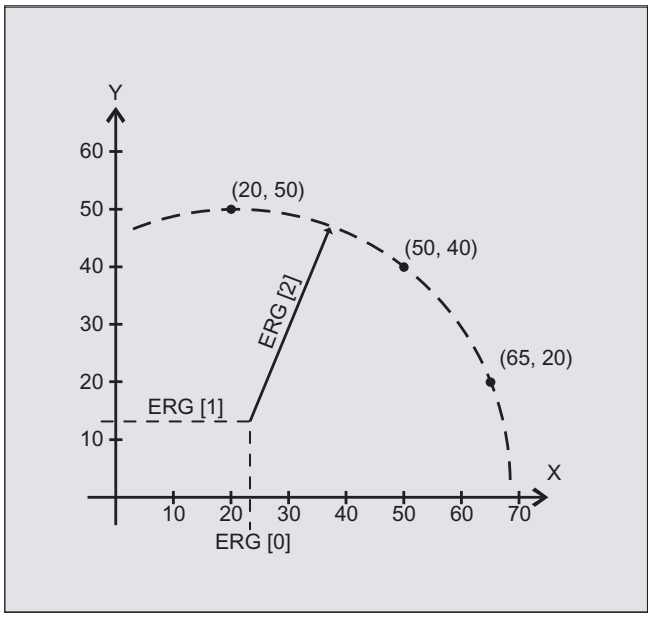
`CALCDAT`	Comando para cálculo do raio e das coordenadas do centro de um círculo a partir de 3 ou 4 pontos		
`<estado>`	Variável para o estado do cálculo do círculo		
	Tipo:	BOOL	
	Valor:	TRUE	Os pontos indicados estão em um círculo.
		FALSE	os pontos indicados **não** estão em um círculo.
`<pontos do círculo>[]`	Variável para indicação dos pontos do círculo com os parâmetros:		
	`<número>`	Número de pontos do círculo (3 ou 4)	
	`<tipo>`	Tipo da indicação de coordenada, p. ex. 2 para coordenadas de 2 pontos	
`<número>`	Parâmetro para o número de pontos utilizados no cálculo (3 ou 4)		
`<resultado>[3]`	Variável para resultado:		
	Indicação de coordenadas do centro do círculo e raio		
	0	Coordenada do centro do círculo: Valor da abscissa	
	1	Coordenada do centro do círculo: Valor da ordenada	
	2	Raio	

Indicação

Observe que as variáveis devem ser definidas antes de sua utilização.

Exemplo

A partir de três pontos deve-se calcular se estes encontram-se em um segmento do círculo.



Código de programa	**Comentário**
`N10 DEF REAL PKT[3,2]=(20,50,50,40,65,20)`	`; Variável para indicação dos pontos do círculo`
`N20 DEF REAL RES[3]`	`; Variável para resultado`
`N30 DEF BOOL STATUS`	`; Variável para estado`
`N40 STATUS=CALCDAT(PTS,3,RES)`	`; Chamada dos dados de círculos calculados.`
`N50 IF STATUS == FALSE GOTOF ERROR`	`; Salto para o erro`

15.7 Desativar a preparação de contorno (EXECUTE)

Função

Com o comando `EXECUTE` desativa-se a preparação de contorno e simultaneamente comuta-se para o modo normal de execução.

Sintaxe

```
EXECUTE(<ERRO>)
```

Significado

`EXECUTE`	Comando para finalização da preparação de contorno
`<ERRO>`	Variável para resposta de erro Tipo: INT O valor das variáveis indica se o contorno pode ser preparado sem erros: 0 Erro 1 Nenhum erro

Exemplo

Código de programa

```
...
N30 CONTPRON(...)
N40 G1 X... Z...
...
N100 EXECUTE(...)
...
```

Tabelas

16

16.1 Lista de instruções

Legenda:

1) Referência para o documento que contém a descrição completa da instrução:

PGsl	Manual de Fundamentos de Programação
PGAsl	Manual de Programação Avançada
BHDsl	Manual de Operação para Torneamento
BHFsl	Manual de operação para Fresamento
FB1 ( )	Manual de funções básicas (com a abreviação alfanumérica da respectiva descrição de funcionamento entre parênteses)
FB2 ( )	Manual de funções ampliadas (com a abreviação alfanumérica da respectiva descrição de funcionamento entre parênteses)
FB3 ( )	Manual de funções especiais (com a abreviação alfanumérica da respectiva descrição de funcionamento entre parênteses)
FBSIsl	Manual de funções para Safety Integrated
FBSY	Manual de funções para ações síncronas
FBW	Manual de funções para gerenciamento de ferramentas

2) Efeito da instrução:

m	modal
b	por bloco

3) Disponibilidade no SINUMERIK 828D (D = torneamento, F = fresamento):

•	Standard
○	Opção
-	Não disponível

4) Ajuste padrão no início do programa (versão de comando fornecida de fábrica, se não houver nada diferente programado).

Instrução	Significado	Para descrição veja 1)	W 2)	828D 3)			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
:	Número de bloco principal NC, fechamento dos marcadores de salto, operador de concatenação	*PGAsl* Funções de cálculo (Página 64)		•	•	•	•
*	Operador para multiplicação	*PGAsl* Funções de cálculo (Página 64)		•	•	•	•

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
+	Operador para adição	*PGAs/* Funções de cálculo (Página 64)		●	●	●	●
-	Operador para subtração	*PGAs/* Funções de cálculo (Página 64)		●	●	●	●
<	Operador de comparação, menor	*PGAs/* Funções de cálculo (Página 64)		●	●	●	●
<<	Operador de concatenação para Strings	*PGAs/* Funções de cálculo (Página 64)		●	●	●	●
<=	Operador de comparação, menor igual	*PGAs/* Funções de cálculo (Página 64)		●	●	●	●
=	Operador de atribuição	*PGAs/* Funções de cálculo (Página 64)		●	●	●	●
>=	Operador de comparação, maior igual	*PGAs/* Funções de cálculo (Página 64)		●	●	●	●
/	Operador para divisão	*PGAs/* Funções de cálculo (Página 64)		●	●	●	●
/0 … … /7	O bloco será omitido (1º nível de omissão) O bloco será omitido (8º nível de omissão)	*PGs/*		● ○	● ○	● ○	● ○
A	Nome de eixo	*PGAs/* Programação da orientação da ferramenta (A..., B..., C..., LEAD, TILT) (Página 329)	m/b	●	●	●	●
A2	Orientação da ferramenta: Ângulo RPY ou ângulo euleriano	*PGAs/* Programação da orientação da ferramenta (A..., B..., C..., LEAD, TILT) (Página 329)	b	●	●	●	●
A3	Orientação da ferramenta: Componente de vetor normal de direção/de área	*PGAs/* Programação da orientação da ferramenta (A..., B..., C..., LEAD, TILT) (Página 329)	b	●	●	●	●
A4	Orientação da ferramenta: Vetor normal de área para o início do bloco	*PGAs/* Fresamento de topo (fresamento 3D A4, B4, C4, A5, B5, C5) (Página 335)	b	●	●	●	●
A5	Orientação da ferramenta: Vetor normal de área para o fim do bloco	*PGAs/* Fresamento de topo (fresamento 3D A4, B4, C4, A5, B5, C5) (Página 335)	b	●	●	●	●
ABS	Valor absoluto (quantia)	*PGAs/* Funções de cálculo (Página 64)		●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
AC	Especificação de dimensão absoluta de coordenadas/posições	*PGs/*	b	•	•	•	•
ACC	Controle da atual aceleração axial	*PGs/*	m	•	•	•	•
ACCLIMA	Controle da atual aceleração axial máxima	*PGs/*	m	•	•	•	•
ACN	Especificação de dimensão absoluta para eixos rotativos, aproximação da posição no sentido negativo	*PGs/*	b	•	•	•	•
ACOS	Arco coseno (função trigonométrica)	*PGAs/* Funções de cálculo (Página 64)		•	•	•	•
ACP	Especificação de dimensão absoluta para eixos rotativos, aproximação da posição no sentido positivo	*PGs/*	b	•	•	•	•
ACTBLOCNO	Retorna o atual número de bloco de um bloco de alarme, mesmo se "ocultar atual exibição de blocos" (DISPLOF) estiver ativo!	*PGAs/* Supressão da atual exibição de bloco (DISPLOF, DISPLON, ACTBLOCNO) (Página 168)		•	•	•	•
ADDFRAME	Inclusão e eventual ativação de um Frame medido	*PGAs/, FB1(K2)* Cálculo de Frame a partir de 3 pontos de medição no espaço (MEAFRAME) (Página 303)		•	•	•	•
ADIS	Distância de suavização para funções de trajetória G1, G2, G3, ...	*PGs/*	m	•	•	•	•
ADISPOS	Distância de suavização para avanço rápido G0	*PGs/*	m	•	•	•	•
ADISPOSA	Tamanho da janela de tolerância para IPOBRKA	*PGAs/* Critérios programáveis para fim de movimentos (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA) (Página 278)	m	•	•	•	•
ALF	Ângulo de retração rápida	*PGAs/* Retração rápida do contorno (SETINT LIFTFAST, ALF) (Página 118)	m	•	•	•	•
AMIRROR	Espelhamento programável	*PGs/*	b	•	•	•	•

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
AND	"E" lógico	*PGAs*/ Operadores de comparação e operadores lógicos (Página 67)		•	•	•	•
ANG	Ângulo de sucessão de elementos de contorno	*PGs*/	b	•	•	•	•
AP	Ângulo polar	*PGs*/	m/b	•	•	•	•
APR	Proteção de acesso para leitura / exibição	*PGAs*/ Atributo: Direitos de acesso (APR, APW, APRP, APWP, APRB, APWB) (Página 41)		•	•	•	•
APRB	Direito de acesso para leitura, BTSS	*PGAs*/ Atributo: Direitos de acesso (APR, APW, APRP, APWP, APRB, APWB) (Página 41)		•	•	•	•
APRP	Direito de acesso para leitura, programa de peça	*PGAs*/ Atributo: Direitos de acesso (APR, APW, APRP, APWP, APRB, APWB) (Página 41)		•	•	•	•
APW	Proteção de acesso para gravação	*PGAs*/ Atributo: Direitos de acesso (APR, APW, APRP, APWP, APRB, APWB) (Página 41)		•	•	•	•
APWB	Direito de acesso para gravação, BTSS	*PGAs*/ Atributo: Direitos de acesso (APR, APW, APRP, APWP, APRB, APWB) (Página 41)		•	•	•	•
APWP	Direito de acesso para gravação, programa de peça	*PGAs*/ Atributo: Direitos de acesso (APR, APW, APRP, APWP, APRB, APWB) (Página 41)		•	•	•	•
APX	Definição da proteção de acesso para a execução do elemento de linguagem indicado	*PGAs*/ Redefinição de variáveis de sistema, variáveis de usuário e comandos de linguagem NC (REDEF) (Página 30)		•	•	•	•
AR	Ângulo de abertura	*PGs*/	m/b	•	•	•	•
AROT	Rotação programável	*PGs*/	b	•	•	•	•
AROTS	Rotações de Frame programáveis com ângulos espaciais	*PGs*/	b	•	•	•	•

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
AS	Definição de macro	*PGAs*/ Técnica de macros (DEFINE ... AS) (Página 205)		●	●	●	●
ASCALE	Escala programável	*PGs*/	b	●	●	●	●
ASIN	Função de cálculo, arco seno	*PGAs*/ Funções de cálculo (Página 64)		●	●	●	●
ASPLINE	Akima-Spline	*PGAs*/ Interpolação de Spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) (Página 237)	m	-	○	-	○
ATAN2	Arco tangente 2	*PGAs*/ Funções de cálculo (Página 64)		●	●	●	●
ATOL	tolerância específica de eixo para funções de compressor, suavização de orientação e tipos de suavização	*PGAs*/ Tolerância de contorno/orientação programável (CTOL, OTOL, ATOL) (Página 503)		-	●	-	●
ATRANS	deslocamento aditivo programável	*PGs*/	b	●	●	●	●
AX	Identificador de eixo variável	*PGAs*/ Funções de eixo (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) (Página 681)	m/b	●	●	●	●
AXCTSWE	Avanço de eixos contentores	*PGAs*/ Contentor de eixo (AXCTSWE, AXCTSWED) (Página 689)		-	-	-	-
AXCTSWED	Rotação de contentor de eixo	*PGAs*/ Contentor de eixo (AXCTSWE, AXCTSWED) (Página 689)		-	-	-	-
AXIS	Identificador de eixo, endereço de eixo	*PGAs*/ Definição de variáveis de usuário (DEF) (Página 24)		●	●	●	●
AXNAME	Converte a String de entrada em identificador de eixo	*PGAs*/ Funções de eixo (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) (Página 681)		●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
AXSTRING	Converte a String em número de fuso	*PGAs*/ Funções de eixo (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) (Página 681)		•	•	•	•
AXTOCHAN	Solicita o eixo para um determinado canal. É possível a partir do programa NC e da ação sincronizada.	*PGAs*/ Transferir eixo de outro canal (AXTOCHAN) (Página 129)		•	•	•	•
AXTOSPI	converte o identificador de eixo em um índice de fuso	*PGAs*/ Funções de eixo (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) (Página 681)		•	•	•	•
B	Nome de eixo	*PGAs*/ Programação da orientação da ferramenta (A..., B..., C..., LEAD, TILT) (Página 329)	m/b	•	•	•	•
B2	Orientação da ferramenta: Ângulo RPY ou ângulo euleriano	*PGAs*/ Programação da orientação da ferramenta (A..., B..., C..., LEAD, TILT) (Página 329)	b	•	•	•	•
B3	Orientação da ferramenta: Componente de vetor normal de direção/de área	*PGAs*/ Programação da orientação da ferramenta (A..., B..., C..., LEAD, TILT) (Página 329)	b	•	•	•	•
B4	Orientação da ferramenta: Vetor normal de área para o início do bloco	*PGAs*/ Fresamento de topo (fresamento 3D A4, B4, C4, A5, B5, C5) (Página 335)	b	•	•	•	•
B5	Orientação da ferramenta: Vetor normal de área para o fim do bloco	*PGAs*/ Fresamento de topo (fresamento 3D A4, B4, C4, A5, B5, C5) (Página 335)	b	•	•	•	•
B_AND	"E" por Bits	*PGAs*/ Operadores de comparação e operadores lógicos (Página 67)		•	•	•	•
B_OR	"OU" por Bits	*PGAs*/ Operadores de comparação e operadores lógicos (Página 67)		•	•	•	•
B_NOT	Negação por Bits	*PGAs*/ Operadores de comparação e operadores lógicos (Página 67)		•	•	•	•

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
B_XOR	"OU" exclusivo por Bits	*PGAs*/ Operadores de comparação e operadores lógicos (Página 67)		●	●	●	●
BAUTO	Definição do primeiro segmento Spline através dos 3 pontos seguintes	*PGAs*/ Interpolação de Spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) (Página 237)	m	-	○	-	○
BLOCK	Definição junto com a palavra-chave TO a parte de programa que deve ser executada em uma execução de subrotina indireta	*PGAs*/ Chamada de subrotina indireta com indicação da parte de programa que deve ser executada (CALL BLOCK ... TO ...) (Página 192)		●	●	●	●
BLSYNC	O processamento da rotina de interrupção apenas deve começar com a próxima mudança de blocos	*PGAs*/ Atribuição e partida de rotinas de interrupção (SETINT, PRIO, BLSYNC) (Página 114)		●	●	●	●
BNAT [4]	Transição natural para o primeiro bloco de Spline	*PGAs*/ Interpolação de Spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) (Página 237)	m	-	○	-	○
BOOL	Tipo de dado: Valores lógicos TRUE/FALSE ou 1/0	*PGAs*/ Definição de variáveis de usuário (DEF) (Página 24)		●	●	●	●
BOUND	Controla se o valor está dentro da faixa de valores definida. A igualdade retorna o valor de controle.	*PGAs*/ Mínimo, máximo e área de variáveis (MINVAL, MAXVAL, BOUND) (Página 71)		●	●	●	●
BRISK [4]	Aceleração de trajetória brusca	*PGs*/	m	●	●	●	●
BRISKA	Ativação da aceleração de trajetória brusca para os eixos programados	*PGs*/		●	●	●	●
BSPLINE	B-Spline	*PGAs*/ Interpolação de Spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) (Página 237)	m	-	○	-	○

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
BTAN	Transição tangencial para o primeiro bloco Spline	*PGAs*/ Interpolação de Spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) (Página 237)	m	-	○	-	○
C	Nome de eixo	*PGAs*/ Programação da orientação da ferramenta (A..., B..., C..., LEAD, TILT) (Página 329)	m/b	●	●	●	●
C2	Orientação da ferramenta: Ângulo RPY ou ângulo euleriano	*PGAs*/ Programação da orientação da ferramenta (A..., B..., C..., LEAD, TILT) (Página 329)	b	●	●	●	●
C3	Orientação da ferramenta: Componente de vetor normal de direção/de área	*PGAs*/ Programação da orientação da ferramenta (A..., B..., C..., LEAD, TILT) (Página 329)	b	●	●	●	●
C4	Orientação da ferramenta: Vetor normal de área para o início do bloco	*PGAs*/ Fresamento de topo (fresamento 3D A4, B4, C4, A5, B5, C5) (Página 335)	b	●	●	●	●
C5	Orientação da ferramenta: Vetor normal de área para o fim do bloco	*PGAs*/ Fresamento de topo (fresamento 3D A4, B4, C4, A5, B5, C5) (Página 335)	b	●	●	●	●
CAC	Aproximação absoluta de uma posição	*PGAs*/ Aproximar posições codificadas (CAC, CIC, CDC, CACP, CACN) (Página 235)		●	●	●	●
CACN	O valor armazenado na tabela é aproximado de forma absoluta em sentido negativo	*PGAs*/ Aproximar posições codificadas (CAC, CIC, CDC, CACP, CACN) (Página 235)		●	●	●	●
CACP	O valor armazenado na tabela é aproximado de forma absoluta em sentido positivo	*PGAs*/ Aproximar posições codificadas (CAC, CIC, CDC, CACP, CACN) (Página 235)		●	●	●	●
CALCDAT	Calcula o raio e o centro de um círculo a partir de 3 ou 4 pontos	*PGAs*/ Calcular dados de círculos (CALCDAT) (Página 727)		●	●	●	●
CALCPOSI	Verificação quanto à violação da área de proteção, limite da área de trabalho e limites de software	*PGAs*/ Verificação quanto à violação da área de proteção, limite da área de trabalho e limites de software (CALCPOSI) (Página 227)		●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
CALL	Chamada de subrotina indireta	*PGAs*/ Chamada de subrotinas indireta (CALL) (Página 191)		●	●	●	●
CALLPATH	Caminho programável de localização para chamada de subrotinas	*PGAs*/ Ampliar o caminho de localização na chamada de subrotinas (CALLPATH) (Página 196)		●	●	●	●
CANCEL	Cancelamento de ação síncrona modal	*PGAs*/ Cancelar ação sincronizada (CANCEL) (Página 648)		●	●	●	●
CASE	Bifurcação de programa condicionada	*PGAs*/ Bifurcação do programa (CASE ... OF ... DEFAULT ...) (Página 89)		●	●	●	●
CDC	Aproximação direta de uma posição	*PGAs*/ Aproximar posições codificadas (CAC, CIC, CDC, CACP, CACN) (Página 235)		●	●	●	●
CDOF [4]	Monitoração de colisão OFF	*PGs*/	m	●	●	●	●
CDOF2	Monitoração de colisão OFF, para fresamento periférico 3D	*PGs*/	m	●	●	●	●
CDON	Monitoração de colisão ON	*PGs*/	m	●	●	●	●
CFC [4]	Avanço constante no contorno	*PGs*/	m	●	●	●	●
CFIN	Avanço constante somente para curvaturas internas, não para curvaturas externas	*PGs*/	m	●	●	●	●
CFINE	Atribuição do deslocamento fino em uma variável FRAME	*PGAs*/ Deslocamento aproximado e deslocamento fino (CFINE, CTRANS) (Página 298)		●	●	●	●
CFTCP	Avanço constante no ponto de referência de corte da ferramenta, trajetória do centro	*PGs*/	m	●	●	●	●
CHAN	Especificação da área de validade de dados	*PGAs*/ Definição de variáveis de usuário (DEF) (Página 24)		●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
CHANDATA	Ajuste do número de canal para acesso de dados no canal	*PGAs*/ Memória de trabalho (CHANDATA, COMPLETE, INITIAL) (Página 214)		●	●	●	●
CHAR	Tipo de dado: Caracteres ASCII	*PGAs*/ Definição de variáveis de usuário (DEF) (Página 24)		●	●	●	●
CHECKSUM	Forma o checksum sobre um campo como STRING de comprimento definido	*PGAs*/ Cálculo de checksum através de um campo (CHECKSUM) (Página 146)		●	●	●	●
CHF	Chanfro; valor = comprimento do chanfro	*PGs*/	b	●	●	●	●
CHKDM	Controle da condição inequívoca dentro de um magazine	*FBW*		●	●	●	●
CHKDNO	Verificação de condição inequívoca de números D	*PGAs*/ Atribuição livre de números D: Verificar números D (CHKDNO) (Página 441)		●	●	●	●
CHR	Chanfro; valor = comprimento do chanfro no sentido de movimento	*PGs*/		●	●	●	●
CIC	Aproximação incremental de uma posição	*PGAs*/ Aproximar posições codificadas (CAC, CIC, CDC, CACP, CACN) (Página 235)		●	●	●	●
CIP	Interpolação circular através do ponto intermediário	*PGs*/	m	●	●	●	●
CLEARM	Resetamento de um ou vários marcadores para coordenação de canal	*PGAs*/ Coordenação de programa (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM) (Página 106)		-	-	-	-
CLRINT	Cancelamento de Interrupt	*PGAs*/ Deletar a atribuição de uma rotina de interrupção (CLRINT) (Página 117)		●	●	●	●
CMIRROR	Espelhamento em um eixo de coordenadas	*PGAs*/ Funções de cálculo (Página 64)		●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
COARSEA	Fim de movimento ao alcançar a "Parada exata aproximada"	*PGAs*/ Critérios programáveis para fim de movimentos (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA) (Página 278)	m	●	●	●	●
COMPCAD	Compressor ON: Qualidade superficial otimizada com programas CAD	*PGAs*/ Compressão de blocos NC (COMPON, COMPCURV, COMPCAD, COMPOF) (Página 251)	m	-	○	-	○
COMPCURV	Compressor ON: polinômio com curvatura contínua	*PGAs*/ Compressão de blocos NC (COMPON, COMPCURV, COMPCAD, COMPOF) (Página 251)	m	-	○	-	○
COMPLETE	Instrução do comando para a saída e entrada de dados	PGAsl Memória de trabalho (CHANDATA, COMPLETE, INITIAL) (Página 214)		●	●	●	●
COMPOF [4]	Compressor OFF	*PGAs*/ Compressão de blocos NC (COMPON, COMPCURV, COMPCAD, COMPOF) (Página 251)	m	-	○	-	○
COMPON	Compressor ON	*PGAs*/ Compressão de blocos NC (COMPON, COMPCURV, COMPCAD, COMPOF) (Página 251)		-	○	-	○
CONTDCON	Decodificação de contorno em formato de tabela ON	*PGAs*/ Criar tabela de contorno codificada (CONTDCON) (Página 720)		●	●	●	●
CONTPRON	Ativação da preparação de referência	*PGAs*/ Criar tabela de contorno (CONTPRON) (Página 714)		●	●	●	●
CORROF	Todos os movimentos sobrepostos ativos são cancelados.	*PGs*/		●	●	●	●
COS	Coseno (função trigonométrica)	*PGAs*/ Funções de cálculo (Página 64)		●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
COUPDEF	Definição do grupo ELG / grupo de fusos síncronos	*PGAs*/ Fuso sincronizado: Programação (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC) (Página 547)		○	-	○	-
COUPDEL	Deletação do grupo ELG	*PGAs*/ Fuso sincronizado: Programação (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC) (Página 547)		○	-	○	-
COUPOF	Grupo ELG / Par de fusos sincronizados ON	*PGAs*/ Fuso sincronizado: Programação (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC) (Página 547)		○	-	○	-
COUPOFS	Desativação do grupo ELG / par de fusos síncronos com parada do fuso escravo	*PGAs*/ Fuso sincronizado: Programação (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC) (Página 547)		○	-	○	-
COUPON	Grupo ELG / Par de fusos sincronizados ON	*PGAs*/ Fuso sincronizado: Programação (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC) (Página 547)		○	-	○	-
COUPONC	Aceitação da ativação do grupo ELG / par de fusos síncronos com programação precedente	*PGAs*/ Fuso sincronizado: Programação (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC) (Página 547)		○	-	○	-
COUPRES	Resetamento do grupo ELG	*PGAs*/ Fuso sincronizado: Programação (COUPDEF, COUPDEL, COUPON, COUPONC, COUPOF, COUPOFS, COUPRES, WAITC) (Página 547)		○	-	○	-

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
CP	Movimento de percurso	*PGAs/* Deslocamento PTP cartesiano (Página 385)	m	●	●	●	●
CPRECOF [4]	Precisão de contorno programável OFF	*PGs/*	m	●	●	●	●
CPRECON	Precisão de contorno programável ON	*PGs/*	m	●	●	●	●
CPROT	Área de proteção específica de canal ON/OFF	*PGAs/* Ativar/desativar áreas de proteção (CPROT, NPROT) (Página 223)		●	●	●	●
CPROTDEF	Definição de uma área de proteção específica de canal	*PGAs/* Definição das áreas de proteção (CPROTDEF, NPROTDEF) (Página 219)		●	●	●	●
CR	Raio do círculo	*PGs/*	b	●	●	●	●
CROT	Rotação do atual sistema de coordenadas	*PGAs/* Funções de cálculo (Página 64)		●	●	●	●
CROTS	Rotações de Frames programáveis com ângulos espaciais (rotações nos eixos especificados)	*PGs/*	b	●	●	●	●
CRPL	Rotação de Frame em um plano qualquer	*FB1(K2)*		●	●	●	●
CSCALE	Fator de escala para vários eixos	*PGAs/* Funções de cálculo (Página 64)		●	●	●	●
CSPLINE	Spline cúbica	*PGAs/* Interpolação de Spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) (Página 237)	m	-	○	-	○
CT	Círculo com transição tangencial	*PGs/*	m	●	●	●	●
CTAB	Posição de eixo escravo determinada com base na posição do eixo mestre a partir da tabela de curvas	*PGAs/* Leitura dos valores da tabela de curvas (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX) (Página 525)		-	-	-	-

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
CTABDEF	Definição de tabela ON	*PGAs*/ Definição de tabelas de curvas (CTABDEF, CATBEND) (Página 514)		-	-	-	-
CTABDEL	Eliminação de tabela de curvas	*PGAs*/ Eliminação de tabelas de curvas (CTABDEL) (Página 520)		-	-	-	-
CTABEND	Definição de tabela OFF	*PGAs*/ Definição de tabelas de curvas (CTABDEF, CATBEND) (Página 514)		-	-	-	-
CTABEXISTS	Verifica a tabela de curva de número n	*PGAs*/ Controle de presença de uma tabela de curvas (CTABEXISTS) (Página 520)		-	-	-	-
CTABFNO	Número de tabelas de curvas possíveis na memória	*PGAs*/ Tabelas de curvas: Controle do aproveitamento de recursos (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL) (Página 530)		-	-	-	-
CTABFPOL	Número de polinômios possíveis na memória	*PGAs*/ Tabelas de curvas: Controle do aproveitamento de recursos (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL) (Página 530)		-	-	-	-
CTABFSEG	Número de segmentos de curva possíveis na memória	*PGAs*/ Tabelas de curvas: Controle do aproveitamento de recursos (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL) (Página 530)		-	-	-	-
CTABID	Fornece o número de tabela da tabela de curvas n	*PGAs*/ Tabelas de curvas: Determinação de propriedades de tabela (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD) (Página 523)		-	-	-	-

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
CTABINV	Posição de eixo mestre determinada com base na posição do eixo escravo a partir da tabela de curvas	*PGAs*/ Leitura dos valores da tabela de curvas (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX) (Página 525)		-	-	-	-
CTABISLOCK	Retorna o estado de bloqueio da tabela de curvas de número n	*PGAs*/ Tabelas de curvas: Determinação de propriedades de tabela (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD) (Página 523)		-	-	-	-
CTABLOCK	Eliminação e sobregravação, bloqueio	*PGAs*/ Bloqueio de tabelas de curvas contra eliminação e sobregravação (CTABLOCK, CTABUNLOCK) (Página 522)		-	-	-	-
CTABMEMTYP	Retorna a memória em que está armazenada a tabela de curva de número n.	*PGAs*/ Tabelas de curvas: Determinação de propriedades de tabela (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD) (Página 523)		-	-	-	-
CTABMPOL	Número máximo de polinômios possíveis na memória	*PGAs*/ Tabelas de curvas: Controle do aproveitamento de recursos (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL) (Página 530)		-	-	-	-
CTABMSEG	Número máximo de segmentos de curva possíveis na memória	*PGAs*/ Tabelas de curvas: Controle do aproveitamento de recursos (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL) (Página 530)		-	-	-	-
CTABNO	Número de tabelas de curvas definidas na SRAM ou DRAM	*FB3(M3)*		-	-	-	-

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
CTABNOMEM	Número de tabelas de curvas definidas na SRAM ou DRAM	*PGAsl* Tabelas de curvas: Controle do aproveitamento de recursos (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL) (Página 530)		-	-	-	-
CTABPERIOD	Retorna a periodicidade de tabela da tabela de curvas de número n	*PGAsl* Tabelas de curvas: Determinação de propriedades de tabela (CTABID, CTABISLOCK, CTABMEMTYP, CTABPERIOD) (Página 523)		-	-	-	-
CTABPOL	Número de polinômios efetivamente utilizados na memória	*PGAsl* Tabelas de curvas: Controle do aproveitamento de recursos (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL) (Página 530)		-	-	-	-
CTABPOLID	Número de polinômios de curvas utilizados pela tabela de curva de número n	*PGAsl* Tabelas de curvas: Controle do aproveitamento de recursos (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL) (Página 530)		-	-	-	-
CTABSEG	Número de segmentos de curva efetivamente utilizados na memória	*PGAsl* Tabelas de curvas: Controle do aproveitamento de recursos (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL) (Página 530)		-	-	-	-

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
CTABSEGID	Número de segmentos de curva utilizados na tabela de curva de número n	*PGAs*/ Tabelas de curvas: Controle do aproveitamento de recursos (CTABNO, CTABNOMEM, CTABFNO, CTABSEGID, CTABSEG, CTABFSEG, CTABMSEG, CTABPOLID, CTABPOL, CTABFPOL, CTABMPOL) (Página 530)		-	-	-	-
CTABSEV	Fornece o valor final do eixo escravo de um segmento da tabela de curva	*PGAs*/ Leitura dos valores da tabela de curvas (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX) (Página 525)		-	-	-	-
CTABSSV	Fornece o valor inicial do eixo escravo de um segmento da tabela de curvas	*PGAs*/ Leitura dos valores da tabela de curvas (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX) (Página 525)		-	-	-	-
CTABTEP	Fornece o valor do eixo mestre no fim da tabela de curvas	*PGAs*/ Leitura dos valores da tabela de curvas (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX) (Página 525)		-	-	-	-
CTABTEV	Fornece o valor do eixo escravo no fim da tabela de curvas	*PGAs*/ Leitura dos valores da tabela de curvas (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX) (Página 525)		-	-	-	-
CTABTMAX	Fornece o valor máximo do eixo escravo da tabela de curvas	*PGAs*/ Leitura dos valores da tabela de curvas (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX) (Página 525)		-	-	-	-

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
CTABTMIN	Fornece o valor mínimo do eixo escravo da tabela de curvas	*PGAs*/ Leitura dos valores da tabela de curvas (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX) (Página 525)		-	-	-	-
CTABTSP	Fornece o valor do eixo mestre no início da tabela de curvas	*PGAs*/ Leitura dos valores da tabela de curvas (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX) (Página 525)		-	-	-	-
CTABTSV	Fornece o valor do eixo escravo no início da tabela de curvas	*PGAs*/ Leitura dos valores da tabela de curvas (CTABTSV, CTABTEV, CTABTSP, CTABTEP, CTABSSV, CTABSEV, CTAB, CTABINV, CTABTMIN, CTABTMAX) (Página 525)		-	-	-	-
CTABUNLOCK	Cancelamento do bloqueio contra eliminação e sobregravação	*PGAs*/ Bloqueio de tabelas de curvas contra eliminação e sobregravação (CTABLOCK, CTABUNLOCK) (Página 522)		-	-	-	-
CTOL	Tolerância de contorno para funções de compressor, suavização de orientação e tipos de suavização	*PGAs*/ Tolerância de contorno/orientação programável (CTOL, OTOL, ATOL) (Página 503)		-	○	-	○
CTRANS	Deslocamento de ponto zero para vários eixos	*PGAs*/ Deslocamento aproximado e deslocamento fino (CFINE, CTRANS) (Página 298)		●	●	●	●
CUT2D [4]	Corretores de ferramenta 2D	*PGs*/	m	●	●	●	●
CUT2DF	Corretores de ferramenta 2D. A correção de ferramenta atua de forma relativa ao atual Frame (plano inclinado).	*PGs*/	m	●	●	●	●
CUT3DC	Corretores de ferramenta 3D no fresamento periférico	*PGAs*/ Ativação da correção de ferramenta 3D (CUT3DC..., CUT3DF...) (Página 420)	m	-	-	-	-

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
CUT3DCC	Corretores de ferramenta 3D no fresamento periférico com superfícies de limitação	*PGAs*/ Correções de ferramenta 3D: Consideração de uma superfície de limitação (CUT3DCC, CUT3DCCD) (Página 430)	m	-	-	-	-
CUT3DCCD	Corretores de ferramenta 3D no fresamento periférico com superfícies de limitação com ferramenta diferencial	*PGAs*/ Correções de ferramenta 3D: Consideração de uma superfície de limitação (CUT3DCC, CUT3DCCD) (Página 430)	m	-	-	-	-
CUT3DF	Correções de ferramenta 3D no fresamento de topo	*PGAs*/ Ativação da correção de ferramenta 3D (CUT3DC..., CUT3DF...) (Página 420)	m	-	-	-	-
CUT3DFF	Correção de ferramenta 3D no fresamento de topo com orientação de ferramenta constante dependente do Frame ativo	*PGAs*/ Ativação da correção de ferramenta 3D (CUT3DC..., CUT3DF...) (Página 420)	m	-	-	-	-
CUT3DFS	Correção de ferramenta 3D no fresamento de topo com orientação de ferramenta constante independente do Frame ativo	*PGAs*/ Ativação da correção de ferramenta 3D (CUT3DC..., CUT3DF...) (Página 420)	m	-	-	-	-
CUTCONOF [4]	Correção de raio constante OFF	*PGs*/	m	●	●	●	●
CUTCONON	Correção de raio constante ON	*PGs*/	m	●	●	●	●
CUTMOD	Ativação da função "Modificação dos dados de corretores para ferramentas orientáveis"	*PGAs*/ Modificação dos dados de corte para ferramentas orientáveis (CUTMOD) (Página 458)		●	●	●	●
CYCLE...	Ciclos de medição	*BHDs*//*BHFs*/					

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
D	Número de corretor da ferramenta	*PGs*/		●	●	●	●
D0	Com D0 as correções da ferramenta estão inativas	*PGs*/		●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
DAC	Programação de diâmetro específica de eixo, absoluta e por blocos	*PGs/*	b	•	•	•	•
DC	Dimensão absoluta para eixos rotativos, aproximação direta da posição	*PGs/*	b	•	•	•	•
DEF	Definição de variáveis	*PGAs/* Definição de variáveis de usuário (DEF) (Página 24)		•	•	•	•
DEFINE	Palavra-chave para definições de macro	*PGAs/* Técnica de macros (DEFINE ... AS) (Página 205)		•	•	•	•
DEFAULT	Deriva na bifurcação CASE	*PGAs/* Bifurcação do programa (CASE ... OF ... DEFAULT ...) (Página 89)		•	•	•	•
DELAYFSTON	Definição do início de uma área Stop-Delay	*PGAs/* Bifurcação do programa (CASE ... OF ... DEFAULT ...) (Página 89)	m	•	•	•	•
DELAYFSTOF	Definição do fim de uma área Stop-Delay	*PGAs/* Segmentos de programa interrompidos condicionalmente (DELAYFSTON, DELAYFSTOF) (Página 480)	m	•	•	•	•
DELDL	Eliminação de correções aditivas	*PGAs/* Deletar correções aditivas (DELDL) (Página 406)		•	•	•	•
DELDTG	Anulação de curso restante	*PGAs/* Anular curso restante (DELDTG) (Página 594)		•	•	•	•
DELETE	Deleta o arquivo especificado. O nome do arquivo pode ser especificado com caminho e extensão de arquivo.	*PGAs/* Deletar arquivo (DELETE) (Página 135)		•	•	•	•
DELTOOLENV	Eliminação de blocos de dados para decrição de ambientes de ferramentas	*FB1(W1)*		•	•	•	•
DIACYCOFA	Programação em diâmetro específica de eixo ativa modalmente: OFF em ciclos	*FB1(P1)*	m	•	•	•	•

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
DIAM90	Programação em diâmetro para G90, programação em raio para G91	*PGAs/*	m	●	●	●	●
DIAM90A	Programação em diâmetro específica de eixo ativa modalmente para G90 e AC, e em raio para G91 e IC	*PGs/*	m	●	●	●	●
DIAMCHAN	Aceitação de todos eixos a partir de funções de eixo de dado de máquina no estado de canal da programação em diâmetro	*PGs/*		●	●	●	●
DIAMCHANA	Aceitação do estado de canal da programação em diâmetro	*PGs/*		●	●	●	●
DIAMCYCOF	Programação em diâmetro específica de canal: OFF em ciclos	*FB1(P1)*	m	●	●	●	●
DIAMOF [4]	Programação em diâmetro: OFF Para posição inicial, veja as informações do fabricante da máquina	*PGs/*	m	●	●	●	●
DIAMOFA	Programação em diâmetro modal e específica de eixo: OFF Para posição inicial, veja as informações do fabricante da máquina	*PGs/*	m	●	●	●	●
DIAMON	Programação em diâmetro: ON	*PGs/*	m	●	●	●	●
DIAMONA	Programação em diâmetro modal e específica de eixo: ON Para habilitação, veja as informações do fabricante da máquina	*PGs/*	m	●	●	●	●
DIC	Programação em diâmetro específica de eixo, relativa e por blocos	*PGs/*	b	●	●	●	●
DILF	Curso de retrocesso (comprimento)	*PGs/*	m	●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
DISABLE	Interrupt OFF	*PGAs/* Desativar / ativar a atribuição de uma rotina de interrupção (DISABLE, ENABLE) (Página 116)		●	●	●	●
DISC	Aceleração do círculo de transição, compensação do raio da ferramenta	*PGs/*	m	●	●	●	●
DISCL	Distância do ponto final do movimento de penetração rápido a partir do plano de usinagem	*PGs/*		●	●	●	●
DISPLOF	Supressão da atual exibição de blocos	*PGAs/* Supressão da atual exibição de bloco (DISPLOF, DISPLON, ACTBLOCNO) (Página 168)		●	●	●	●
DISPLON	Cancelamento da supressão da atual exibição de blocos	*PGAs/* Supressão da atual exibição de bloco (DISPLOF, DISPLON, ACTBLOCNO) (Página 168)		●	●	●	●
DISPR	Diferença de percurso de reposicionamento	*PGAs/* Reaproximação no contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN) (Página 488)	b	●	●	●	●
DISR	Distância de reposicionamento	*PGAs/* Reaproximação no contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN) (Página 488)	b	●	●	●	●
DITE	Curso de saída da rosca	*PGs/*	m	●	●	●	●
DITS	Curso de entrada da rosca	*PGs/*	m	●	●	●	●
DIV	Divisão Integer	*PGAs/* Funções de cálculo (Página 64)		●	●	●	●
DL	Seleção de corretor de ferramenta aditivo dependente de local (DL, corretor aditivo, corretor de ajuste)	*PGAs/* Ativar correções aditivas (DL) (Página 403)	m	-	-	-	-

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
DO	Palavra-chave para ação síncrona, ativa a ação quando a condição for preenchida.	*PGAs*/ Ações (DO) (Página 569)		●	●	●	●
DRFOF	Desativação dos deslocamentos com manivela eletrônica (DRF)	*PGs*/	m	●	●	●	●
DRIVE	Aceleração de trajetória em função da velocidade	*PGs*/	m	●	●	●	●
DRIVEA	Ativação da curva característica de aceleração dobrada para os eixos programados	*PGs*/		●	●	●	●
DYNFINISH	Dinâmica para acabamento fino	*PGs*/	m	●	●	●	●
DYNNORM	Dinâmica normal	*PGs*/	m	●	●	●	●
DYNPOS	Dinâmica para modo de posicionamento, rosqueamento com macho	*PGs*/	m	●	●	●	●
DYNROUGH	Dinâmica para desbaste	*PGs*/	m	●	●	●	●
DYNSEMIFIN	Dinâmica para acabamento	*PGs*/	m	●	●	●	●
DZERO	Marca todos números D da unidade TO como inválidos	*PGAs*/ Atribuição livre de números D: Invalidar números D (DZERO) (Página 444)		●	●	●	●
EAUTO	Definição do último segmento Spline através dos últimos 3 pontos	*PGAs*/ Interpolação de Spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) (Página 237)	m	-	○	-	○
EGDEF	Definição de uma caixa de transmissão eletrônica	*PGAs*/ Definir caixa de transmissão eletrônica (EGDEF) (Página 538)		-	-	-	-
EGDEL	Eliminação da definição de acoplamento para o eixo escravo	*PGAs*/ Deletar a definição de uma caixa de transmissão eletrônica (EGDEL) (Página 544)		-	-	-	-

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
EGOFC	Desativação contínua da caixa de transmissão eletrônica	*PGAs*/ Desativar transmissão eletrônica (EGOFS, EGOFC) (Página 543)		-	-	-	-
EGOFS	Desativação seletiva da caixa de transmissão eletrônica	*PGAs*/ Desativar transmissão eletrônica (EGOFS, EGOFC) (Página 543)		-	-	-	-
EGON	Ativação da caixa de transmissão eletrônica	*PGAs*/ Desativar transmissão eletrônica (EGOFS, EGOFC) (Página 543)		-	-	-	-
EGONSYN	Ativação da caixa de transmissão eletrônica	*PGAs*/ Desativar transmissão eletrônica (EGOFS, EGOFC) (Página 543)		-	-	-	-
EGONSYNE	Ativação da caixa de transmissão eletrônica, com especificação do modo de aproximação	*PGAs*/ Desativar transmissão eletrônica (EGOFS, EGOFC) (Página 543)		-	-	-	-
ELSE	Bifurcação do programa, se a condição IF não for preenchida	*PGAs*/ Loop de programa com alternativa (IF, ELSE, ENDIF) (Página 99)		●	●	●	●
ENABLE	Interrupt ON	*PGAs*/ Desativar / ativar a atribuição de uma rotina de interrupção (DISABLE, ENABLE) (Página 116)		●	●	●	●
ENAT [4]	Transição de curvas natural para o próximo bloco de deslocamento	*PGAs*/ Interpolação de Spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) (Página 237)	m	-	○	-	○
ENDFOR	Linha final do loop de contagem FOR	*PGAs*/ Loop contador (FOR ... TO ..., ENDFOR) (Página 102)		●	●	●	●
ENDIF	Linha final da bifurcação IF	*PGAs*/ Loop de programa com alternativa (IF, ELSE, ENDIF) (Página 99)		●	●	●	●
ENDLABEL	Marcador final para repetições de programa de peça através do REPEAT	*PGAs, FB1(K1)* Repetição de partes do programa (REPEAT, REPEATB, ENDLABEL, P) (Página 91)		●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
ENDLOOP	Linha final do loop contínuo de programa LOOP	*PGAs*/ Loop de programa infinito (LOOP, ENDLOOP) (Página 101)		●	●	●	●
ENDPROC	Linha final de um programa com linha inicial PROC			●	●	●	●
ENDWHILE	Linha final do loop WHILE	*PGAs*/ Loop de programa com condição no início do loop (WHILE, ENDWHILE) (Página 103)		●	●	●	●
ETAN	Transição tangencial de curva para o próximo bloco de deslocamento no início de Spline	*PGAs*/ Interpolação de Spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) (Página 237)	m	-	○	-	○
EVERY	Execução de ação síncrona na passagem da condição de FALSE para TRUE	*PGAs*/ Controle cíclico da condição (WHEN, WHENEVER, FROM, EVERY) (Página 567)		●	●	●	●
EX	Palavra-chave para a atribuição de valor na forma escrita exponencial	*PGAs*/ Variáveis de usuário pré-definidas: Parâmetros de cálculo (R) (Página 20)		●	●	●	●
EXECSTRING	Transferência de uma variável String com a linha de programa de peça a ser executada	*PGAs*/ Programação indireta de linhas de programa de peça (EXECSTRING) (Página 63)		●	●	●	●
EXECTAB	Execução de um elemento a partir de uma tabela de movimentos	*PGAs*/ Programação indireta de linhas de programa de peça (EXECSTRING) (Página 63)		●	●	●	●
EXECUTE	Execução de programa ON	*PGAs*/ Desativar a preparação de contorno (EXECUTE) (Página 729)		●	●	●	●
EXP	Função exponencial ex	*PGAs*/ Funções de cálculo (Página 64)		●	●	●	●
EXTCALL	Execução de subrotina externa	*PGAs*/ Executar subrotina externa (EXTCALL) (Página 197)		●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
EXTERN	Identificação de uma subrotina com transferência de parâmetros	*PGAs*/ Chamada de subrotina sem transferência de parâmetros (Página 182)		•	•	•	•
F	Valor de avanço (em associação com G4 também é programado o tempo de espera com o F)	*PGs*/		•	•	•	•
FA	Avanço axial	*PGs*/	m	•	•	•	•
FAD	Avanço de penetração para aproximação suave e afastamento suave	*PGs*/		•	•	•	•
FALSE	Constante lógica: incorreto	*PGAs*/ Definição de variáveis de usuário (DEF) (Página 24)		•	•	•	•
FB	Avanço por bloco	*PGs*/		•	•	•	•
FCTDEF	Definição de função polinomial	*PGAs*/ Correção de ferramenta Online (PUTFTOCF, FCTDEF, PUTFTOC, FTOCON, FTOCOF) (Página 415)		-	-	-	-
FCUB	Avanço variável conforme Spline cúbica	*PGAs*/ Característica de avanço (FNORM, FLIN, FCUB, FPO) (Página 472)	m	•	•	•	•
FD	Avanço de trajetória para sobreposição com manivela eletrônica	*PGs*/	b	•	•	•	•
FDA	Avanço axial para sobreposição de manivela eletrônica	*PGs*/	b	•	•	•	•
FENDNORM	Desaceleração de cantos OFF	*PGAs*/ Redução de avanço com desaceleração nos cantos (FENDNORM, G62, G621) (Página 277)	m	•	•	•	•
FFWOF [4]	Controle feedforward OFF	*PGs*/	m	•	•	•	•
FFWON	Controle feedforward ativado	*PGs*/	m	•	•	•	•

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
FGREF	Raio de referência em eixos rotativos ou fatores de referência de trajetória em eixos de orientação (interpolação de vetores)	*PGs/*	m	•	•	•	•
FGROUP	Definição dos eixos com avanço de trajetória	*PGs/*		•	•	•	•
FI	Parâmetro para acesso aos dados de Frame: Deslocamento fino	*PGAs/* Ler e alterar componentes de Frame (TR, FI, RT, SC, MI) (Página 294)		•	•	•	•
FIFOCTRL	Controle da memória de pré-processamento	*PGAs/* Execução do programa com memória de pré-processamento (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE) (Página 477)	m	•	•	•	•
FILEDATE	Retorna a data do último acesso de gravação do arquivo	*PGAs/* Leitura das informações de arquivo (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO) (Página 143)		•	•	•	•
FILEINFO	Retorna a soma de FILEDATE, FILESIZE, FILESTAT e FILETIME juntos	*PGAs/* Leitura das informações de arquivo (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO) (Página 143)		•	•	•	•
FILESIZE	Retorna o tamanho atual do arquivo	*PGAs/* Leitura das informações de arquivo (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO) (Página 143)		•	•	•	•
FILESTAT	Retorna o estado de arquivo dos direitos de leitura, gravação, execução, exibição e deletação (rwxsd)	*PGAs/* Leitura das informações de arquivo (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO) (Página 143)		•	•	•	•
FILETIME	Retorna o horário do último acesso de gravação do arquivo	*PGAs/* Leitura das informações de arquivo (FILEDATE, FILETIME, FILESIZE, FILESTAT, FILEINFO) (Página 143)		•	•	•	•

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
FINEA	Fim de movimento ao alcançar a "Parada exata fina"	*PGAs*/ Critérios programáveis para fim de movimentos (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA) (Página 278)	m	●	●	●	●
FL	Velocidade limite para eixos síncronos	*PGs*/ Ler e alterar componentes de Frame (TR, FI, RT, SC, MI) (Página 294)	m	●	●	●	●
FLIN	Avanço linear variável	*PGAs*/ Característica de avanço (FNORM, FLIN, FCUB, FPO) (Página 472)	m	●	●	●	●
FMA	Vários avanços axiais	*PGs*/	m	-	-	-	-
FNORM [4]	Avanço normal conforme DIN66025	*PGAs*/ Característica de avanço (FNORM, FLIN, FCUB, FPO) (Página 472)	m	●	●	●	●
FOCOF	Desativação do deslocamento com momento/força limitados	*PGAs*/ Deslocar até o encosto fixo (FXS, FXST, FXSW, FOCON, FOCOF) (Página 632)	m	○	-	○	-
FOCON	Ativação do deslocamento com momento/força limitados	*PGAs*/ Deslocar até o encosto fixo (FXS, FXST, FXSW, FOCON, FOCOF) (Página 632)	m	○	-	○	-
FOR	Loop de contagem com número fixo de passadas	*PGAs*/ Loop contador (FOR ... TO ..., ENDFOR) (Página 102)		●	●	●	●
FP	Ponto fixo: Número do ponto fixo a ser aproximado	*PGs*/	b	●	●	●	●
FPO	Caracaterística de avanço programada através de um polinômio	*PGAs*/ Característica de avanço (FNORM, FLIN, FCUB, FPO) (Página 472)		-	-	-	-
FPR	Identificação do eixo rotativo	*PGs*/		●	●	●	●
FPRAOF	Desativação do avanço por rotação	*PGs*/		●	●	●	●
FPRAON	Ativação do avanço por rotação	*PGs*/		●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
FRAME	Tipo de dado para definição de sistemas de coordenadas	*PGAs*/ Definição de novos Frames (DEF FRAME) (Página 297)		●	●	●	●
FRC	Avanço para raio e chanfro	*PGs*/	b	●	●	●	●
FRCM	Avanço para raio e chanfro modal	*PGs*/	m	●	●	●	●
FROM	A ação é executada quando a condição é preenchida uma vez e permanece ativa por toda a ação síncrona	*PGAs*/ Controle cíclico da condição (WHEN, WHENEVER, FROM, EVERY) (Página 567)		●	●	●	●
FTOC	Modificação da correção fina de ferramenta	*PGs*/ Correção de ferramenta Online (PUTFTOCF, FCTDEF, PUTFTOC, FTOCON, FTOCOF) (Página 415)		●	●	●	●
FTOCOF [4]	Correção fina de ferramenta ativa Online OFF	*PGAs*/ Correção de ferramenta Online (PUTFTOCF, FCTDEF, PUTFTOC, FTOCON, FTOCOF) (Página 415)	m	●	●	●	●
FTOCON	Correção fina de ferramenta ativa Online ON	*PGAs*/ Correção de ferramenta Online (PUTFTOCF, FCTDEF, PUTFTOC, FTOCON, FTOCOF) (Página 415)	m	●	●	●	●
FXS	Deslocamento até o encosto fixo ativado	*PGs*/ Deslocar até o encosto fixo (FXS, FXST, FXSW, FOCON, FOCOF) (Página 632)	m	●	●	●	●
FXST	Limite de torque para deslocamento até o encosto fixo	*PGs*/ Deslocar até o encosto fixo (FXS, FXST, FXSW, FOCON, FOCOF) (Página 632)	m	●	●	●	●
FXSW	Janela de monitoração para deslocamento até o encosto fixo	*PGs*/ Deslocar até o encosto fixo (FXS, FXST, FXSW, FOCON, FOCOF) (Página 632)		●	●	●	●
FZ	Avanço por dente	*PGs*/	m	●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
G0	Interpolação linear com avanço (movimento) rápido	*PGs/*	m	•	•	•	•
G1 [4]	Interpolação linear com avanço (interpolação de retas)	*PGs/*	m	•	•	•	•
G2	Interpolação circular em sentido horário	*PGs/*	m	•	•	•	•
G3	Interpolação circular em sentido anti-horário	*PGs/*	m	•	•	•	•
G4	Tempo de espera, pré-definido	*PGs/*	b	•	•	•	•
G5	Retificação inclinada de canal	*PGAs/* Eixo inclinado (TRAANG) (Página 380)	b	•	•	•	•
G7	Movimento de compensação na retificação inclinada de canal	*PGAs/* Eixo inclinado (TRAANG) (Página 380)	b	•	•	•	•
G9	Parada exata - desaceleração	*PGs/*	b	•	•	•	•
G17 [4]	Seleção do plano de trabalho X/Y	*PGs/*	m	•	•	•	•
G18	Seleção do plano de trabalho Z/X	*PGs/*	m	•	•	•	•
G19	Seleção do plano de trabalho Y/Z	*PGs/*	m	•	•	•	•
G25	Limite inferior da área de trabalho	*PGs/*	b	•	•	•	•
G26	Limite da área de trabalho superior	*PGs/*	b	•	•	•	•
G33	Rosqueamento com passo constante	*PGs/*	m	•	•	•	•
G34	Rosqueamento com passo linear e crescente	*PGs/*	m	•	•	•	•
G35	Rosqueamento com passo linear e decrescente	*PGs/*	m	•	•	•	•
G40 [4]	Correção do raio da ferramenta OFF	*PGs/*	m	•	•	•	•
G41	Correção do raio da ferramenta à esquerda do contorno	*PGs/*	m	•	•	•	•

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
G42	Correção do raio da ferramenta à direita do contorno	*PGs/*	m	●	●	●	●
G53	Supressão do atual deslocamento de ponto zero (por bloco)	*PGs/*	b	●	●	●	●
G54	1º deslocamento de ponto zero ajustável	*PGs/*	m	●	●	●	●
G55	2º deslocamento do ponto zero ajustável	*PGs/*	m	●	●	●	●
G56	3º deslocamento de ponto zero ajustável	*PGs/*	m	●	●	●	●
G57	4º deslocamento de ponto zero ajustável	*PGs/*	m	●	●	●	●
G58	Deslocamento de ponto zero axial e programável de modo absoluto, deslocamento aproximado	*PGs/*	b	●	●	●	●
G59	Deslocamento de ponto zero axial e programável de modo aditivo, deslocamento fino	*PGs/*	b	●	●	●	●
G60 [4]	Parada exata - desaceleração	*PGs/*	m	●	●	●	●
G62	Desaceleração em cantos internos com compensação do raio da ferramenta ativa (G41, G42)	*PGAs/* Redução de avanço com desaceleração nos cantos (FENDNORM, G62, G621) (Página 277)	m	●	●	●	●
G63	Rosqueamento com macho com mandril de compensação	*PGs/*	b	●	●	●	●
G64	Modo de controle da trajetória	*PGs/*	m	●	●	●	●
G70	Especificação em polegadas para dimensões geométricas (comprimentos)	*PGs/*	m	●	●	●	●
G71 [4]	Especificação métrica para dimensões geométricas (comprimentos)	*PGs/*	m	●	●	●	●
G74	Aproximação do ponto de referência	*PGs/*	b	●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
G75	Aproximação do ponto fixo	*PGs/*	b	•	•	•	•
G90 [4]	Especificação de dimensão absoluta	*PGs/*	m/b	•	•	•	•
G91	Especificação de dimensão incremental	*PGs/*	m/b	•	•	•	•
G93	Avanço em função do tempo 1/min (rpm)	*PGs/*	m	•	•	•	•
G94 [4]	Avanço linear F em mm/min ou pol./min e graus/min	*PGs/*	m	•	•	•	•
G95	Avanço por rotação F em mm/rot. ou pol./rot.	*PGs/*	m	•	•	•	•
G96	Velocidade de corte constante (como no G95) ON	*PGs/*	m	•	•	•	•
G97	Velocidade de corte constante (como no G95) OFF	*PGs/*	m	•	•	•	•
G110	Programação polar relativa à última posição nominal programada	*PGs/*	b	•	•	•	•
G111	Programação polar relativa ao ponto zero do atual sistema de coordenadas da peça de trabalho	*PGs/*	b	•	•	•	•
G112	Programação polar relativa ao último pólo válido	*PGs/*	b	•	•	•	•
G140 [4]	Sentido de aproximação WAB definido através de G41/G42	*PGs/*	m	•	•	•	•
G141	Sentido de aproximação WAB à esquerda do contorno	*PGs/*	m	•	•	•	•
G142	Sentido de aproximação WAB à direita do contorno	*PGs/*	m	•	•	•	•
G143	Sentido de aproximação WAB em função da tangente	*PGs/*	m	•	•	•	•
G147	Aproximação suave em linha reta	*PGs/*	b	•	•	•	•
G148	Afastamento suave em linha reta	*PGs/*	b	•	•	•	•

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
G153	Supressão do atual Frame inclusive Frame básico	*PGs/*	b	●	●	●	●
G247	Aproximação suave em quadrante	*PGs/*	b	●	●	●	●
G248	Afastamento suave em quadrante	*PGs/*	b	●	●	●	●
G290	Comutação para modo SINUMERIK ON	*FBW*	m	●	●	●	●
G291	Comutação para modo ISO2/3 ON	*FBW*	m	●	●	●	●
G331	Rosqueamento com macho sem mandril de compensação, passo positivo, giro horário (direito)	*PGs/*	m	●	●	●	●
G332	Rosqueamento com macho sem mandril de compensação, passo negativo, giro anti-horário (esquerdo)	*PGs/*	m	●	●	●	●
G340 [4]	Bloco de aproximação espacial (simultâneo em profundidade e no plano (espiral))	*PGs/*	m	●	●	●	●
G341	Primeiro penetração no eixo perpendicular (z), depois aproximação no plano	*PGs/*	m	●	●	●	●
G347	Aproximação suave em semicírculo	*PGs/*	b	●	●	●	●
G348	Afastamento suave em semicírculo	*PGs/*	b	●	●	●	●
G450 [4]	Círculo de transição	*PGs/*	m	●	●	●	●
G451	Intersecção das equidistâncias	*PGs/*	m	●	●	●	●
G460 [4]	Ativação da monitoração de colisão para bloco de aproximação e de afastamento	*PGs/*	m	●	●	●	●
G461	Inserção de um círculo no bloco de compensação do raio de ferramenta (WRK)	*PGs/*	m	●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
G462	Inserção de uma reta no bloco de compensação do raio de ferramenta (WRK)	*PGs/*	m	•	•	•	•
G500 [4]	Desativação de todos os Frames ajustáveis, Frames básicos estão ativos	*PGs/*	m	•	•	•	•
G505 ... G599	5 ... 99. Deslocamento de ponto zero ajustável	*PGs/*	m	•	•	•	•
G601 [4]	Mudança de blocos com parada exata fina	*PGs/*	m	•	•	•	•
G602	Mudança de blocos com parada exata aproximada	*PGs/*	m	•	•	•	•
G603	Mudança de blocos para fim de bloco de interpolação IPO	*PGs/*	m	•	•	•	•
G621	Desaceleração de cantos em todos os cantos	*PGAs/* Redução de avanço com desaceleração nos cantos (FENDNORM, G62, G621) (Página 277)	m	•	•	•	•
G641	Modo de controle da trajetória com suavização conforme critério de percurso (= distância de suavização programável)	*PGs/*	m	•	•	•	•
G642	Modo de controle da trajetória com suavização com a conservação de tolerâncias definidas	*PGs/*	m	•	•	•	•
G643	Modo de controle da trajetória com suavização com a conservação de tolerâncias definidas (interno de bloco)	*PGs/*	m	•	•	•	•
G644	Modo de controle da trajetória com suavização com o máximo possível de dinâmica	*PGs/*	m	•	•	•	•
G645	Modo de controle da trajetória com suavização de cantos e transições de blocos tangenciais com preservação de tolerâncias definidas	*PGs/*	m	•	•	•	•

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
G700	Especificação em polegadas para dimensões geométricas e tecnológicas (comprimentos, avanço)	*PGsl*	m	●	●	●	●
G710 [4]	Especificação métrica para dimensões geométricas e tecnológicas (comprimentos, avanço)	*PGsl*	m	●	●	●	●
G751	Aproximação do ponto fixo através de ponto intermediário	PGsl	b	●	●	●	●
G810 [4], ..., G819	Grupo G reservado para o usuário OEM	*PGAsl* Funções especiais para usuários OEM (OEMIPO1, OEMIPO2, G810 até G829) (Página 276)		●	●	●	●
G820 [4], ..., G829	Grupo G reservado para o usuário OEM	*PGAsl* Funções especiais para usuários OEM (OEMIPO1, OEMIPO2, G810 até G829) (Página 276)		●	●	●	●
G931	Especificação de avanço através do tempo de deslocamento		m	●	●	●	●
G942	Congelamento do avanço linear e velocidade de corte constante ou rotação de fuso		m	●	●	●	●
G952	Congelamento do avanço por rotação e velocidade de corte constante ou rotação de fuso		m	●	●	●	●
G961	velocidade de corte constante e avanço linear	*PGsl*	m	●	●	●	●
G962	Avanço linear ou avanço por rotação e velocidade de corte constante	*PGsl*	m	●	●	●	●
G971	Congelamento da rotação do fuso e avanço linear	*PGsl*	m	●	●	●	●
G972	Congelamento do avanço linear ou avanço por rotação e rotação constante de fuso	*PGsl*	m	●	●	●	●
G973	Avanço de rotação sem limite da rotação do fuso	*PGsl*	m	●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
GEOAX	Atribui novos canais de eixo para os eixos geométricos 1 - 3	*PGAsl* Eixos geométricos comutáveis (GEOAX) (Página 684)		•	•	•	•
GET	Troca de eixos liberados entre canais	*PGAsl* Troca de eixos, troca de fusos (RELEASE, GET, GETD) (Página 124)		•	•	•	•
GETACTT	Determina a ferramenta ativa de um grupo de ferramentas de mesmo nome	*FBW*		•	•	•	•
GETACTTD	Determina para um número D absoluto seu número T correspondente	*PGAsl* Atribuição livre de números D: Determinar o número T para o número D especificado (GETACTTD) (Página 443)		•	•	•	•
GETD	Troca de eixo direta entre canais	*PGAsl* Troca de eixos, troca de fusos (RELEASE, GET, GETD) (Página 124)		•	•	•	•
GETDNO	Fornece o número D de um corte (CE) de uma ferramenta (T)	*PGAsl* Atribuição livre de números D: Renomear números D (GETDNO, SETDNO) (Página 442)		•	•	•	•
GETEXET	Leitura do número T carregado	*FBW*		•	•	•	•
GETFREELOC	Localização de um alojamento vazio no magazine para uma ferramenta especificada	*FBW*		•	•	•	•
GETSELT	Fornecer números T pré-selecionados	*FBW*		•	•	•	•
GETT	Definir número T para nome de ferramenta	*FBW*		•	•	•	•
GETTCOR	Extração de dados de comprimentos de ferramenta ou componentes de comprimento de ferramenta	*FB1(W1)*		•	•	•	•
GETTENV	Leitura de números T, D e DL	*FB1(W1)*		•	•	•	•

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
GOTO	Instrução de salto primeiro para frente depois para trás (sentido primeiro para o fim e depois para o início do programa)	*PGAs/* Saltos de programa até marcadores de salto (GOTOB, GOTOF, GOTO, GOTOC) (Página 86)		•	•	•	•
GOTOB	Instrução de salto para trás (sentido no início do programa)	*PGAs/* Saltos de programa até marcadores de salto (GOTOB, GOTOF, GOTO, GOTOC) (Página 86)		•	•	•	•
GOTOC	Como GOTO, mas com supressão do alarme 14080 "Destino de salto não encontrado"	*PGAs/* Saltos de programa até marcadores de salto (GOTOB, GOTOF, GOTO, GOTOC) (Página 86)		•	•	•	•
GOTOF	Instrução de salto para frente (sentido no fim do programa)	*PGAs/* Saltos de programa até marcadores de salto (GOTOB, GOTOF, GOTO, GOTOC) (Página 86)		•	•	•	•
GOTOS	Salto de retorno ao início do programa	*PGAs/* Salto de retorno ao início do programa (GOTOS) (Página 85)		•	•	•	•
GP	Palavra-chave para programação indireta de atributos de posição	*PGAs/* Programação indireta de atributos de posição (GP) (Página 60)		•	•	•	•
GWPSOF	Cancelamento da velocidade periférica de rebolo constante (SUG)	*PGs/*	b	•	•	•	•
GWPSON	Ativação da velocidade periférica de rebolo constante (SUG)	*PGs/*	b	•	•	•	•
H...	Emissão de função auxiliar no PLC	*PGs/FB1(H2)*		•	•	•	•
HOLES1	Ciclo de modelo de furação, fileira de furos	*BHDs/BHFs/*		•	•	•	•
HOLES2	Ciclo de modelo de furação, círculo de furos	*BHDs/BHFs/*		•	•	•	•
I	Parâmetro de interpolação	*PGs/*	b	•	•	•	•
I1	Coordenada de ponto intermediário	*PGs/*	b	•	•	•	•

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
IC	Especificação de dimensões incrementais	*PGs/*	b	●	●	●	●
ICYCOF	Execução de todos blocos de um ciclo de tecnologia conforme ICYCOF em um ciclo IPO	*PGAs/* Controle da execução de ciclos de tecnologia (ICYCOF, ICYCON) (Página 643)		●	●	●	●
ICYCON	Execução de cada bloco de um ciclo de tecnologia conforme ICYCOF em um ciclo IPO separado	*PGAs/* Controle da execução de ciclos de tecnologia (ICYCOF, ICYCON) (Página 643)		●	●	●	●
ID	Identificação para ações síncronas modais	*PGAs/* Área de validade e seqüência de usinagem (ID, IDS) (Página 565)	m	●	●	●	●
IDS	Identificação para ações síncronas estáticas modais	*PGAs/* Área de validade e seqüência de usinagem (ID, IDS) (Página 565)		●	●	●	●
IF	Introdução de um salto condicional no programa de peça / ciclo de tecnologia	*PGAs/* Loop de programa com alternativa (IF, ELSE, ENDIF) (Página 99)		●	●	●	●
INDEX	Determinação do índice de um caractere na String de entrada	*PGAs/* Localizar caractere/String na String (INDEX, RINDEX, MINDEX, MATCH) (Página 81)		●	●	●	●
INIPO	Inicialização das variáveis com PowerOn	*PGAs/* Definição de variáveis de usuário (DEF) (Página 24)		●	●	●	●
INIRE	Inicialização das variáveis com Reset	*PGAs/* Definição de variáveis de usuário (DEF) (Página 24)		●	●	●	●
INICF	Inicialização das variáveis com NewConfig	*PGAs/* Definição de variáveis de usuário (DEF) (Página 24)		●	●	●	●
INIT	Seleção de um determinado programa NC para execução em um determinado canal	*PGAs/* Coordenação de programa (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM) (Página 106)		-	-	-	-
INITIAL	Criação de um INI-File através de todas as áreas	*PGAs/* Memória de trabalho (CHANDATA, COMPLETE, INITIAL) (Página 214)		●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
INT	Tipo de dado: Valor inteiro com sinal	*PGAs*/ Definição de variáveis de usuário (DEF) (Página 24)		●	●	●	●
INTERSEC	Calcula a intersecção entre dois elementos de contorno	*PGAs*/ Determinar a intersecção entre dois elementos de contorno (INTERSEC). (Página 724)		●	●	●	●
INVCCW	Deslocamento de evolvente, no sentido anti-horário	*PGs*/	m	-	-	-	-
INVCW	Deslocamento de evolvente, no sentido horário	*PGs*/	m	-	-	-	-
INVFRAME	Cálculo do Frame inverso a partir de um Frame	*FB1(K2)*		●	●	●	●
IP	Parâmetro de interpolação variável	*PGAs*/ Programação indireta (Página 56)		●	●	●	●
IPOBRKA	Critério de movimento a partir do ponto de ativação da rampa de frenagem	*PGAs*/ Critérios programáveis para fim de movimentos (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA) (Página 278)	m	●	●	●	●
IPOENDA	Fim de movimento ao alcançar "IPO-Stop"	*PGAs*/ Critérios programáveis para fim de movimentos (FINEA, COARSEA, IPOENDA, IPOBRKA, ADISPOSA) (Página 278)	m	●	●	●	●
IPTRLOCK	Congela o início do segmento do programa que não deve ser pesquisado até o próximo bloco de função da máquina.	*PGAs*/ Evitar posição de programa para SERUPRO (IPTRLOCK, IPTRUNLOCK) (Página 485)	m	●	●	●	●
IPTRUNLOCK	Define o fim do segmento do programa que não deve ser pesquisado no bloco atual no momento da interrupção.	*PGAs*/ Evitar posição de programa para SERUPRO (IPTRLOCK, IPTRUNLOCK) (Página 485)	m	●	●	●	●
ISAXIS	Verifica se o eixo geométrico especificado como parâmetro é 1	*PGAs*/ Funções de eixo (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) (Página 681)		●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
ISD	Profundidade de imersão	*PGAs/* Ativação das correções de ferramentas 3D (CUT3DC, CUT3DF, CUT3DFS, CUT3DFF, ISD) (Página 420)	m	•	•	•	•
ISFILE	Verifica se existe um arquivo na memória de usuário do NCK	*PGAs/* Controle de presença de um arquivo (ISFILE) (Página 141)		•	•	•	•
ISNUMBER	Verifica se a String de entrada pode ser convertida em número	*PGAs/* Conversão de tipos de STRING (NUMBER, ISNUMBER, AXNAME) (Página 77)		•	•	•	•
ISOCALL	Chamada indireta de um programa programado em linguagem ISO	*PGAs/* Chamada indireta de um programa programado em linguagem ISO (ISOCALL) (Página 194)		•	•	•	•
ISVAR	Verifica se o parâmetro de transferência contém uma variável conhecida do NC	*PGAs/* Chamada de função ISVAR e leitura do índice Array de dados de máquina (Página 699)		•	•	•	•
J	Parâmetro de interpolação	*PGs/*	b	•	•	•	•
J1	Coordenada de ponto intermediário	*PGs/*	b	•	•	•	•
JERKA	Ativação do comportamento de aceleração ajustado através de MD para os eixos programados			•	•	•	•
JERKLIM	Redução ou aceleração do solavanco axial máximo	*PGAs/* Correção percentual de solavancos (JERKLIM) (Página 498)	m	•	•	•	•
JERKLIMA	Redução ou aceleração do solavanco axial máximo	*PGs/*	m	•	•	•	•
K	Parâmetro de interpolação	*PGs/*	b	•	•	•	•
K1	Coordenada de ponto intermediário	*PGs/*	b	•	•	•	•
KONT	Percorre o contorno com compensação da ferramenta	*PGs/*	m	•	•	•	•

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
KONTC	Aproximação/afastamento com polinômio de curvatura contínua	*PGs/*	m	●	●	●	●
KONTT	Aproximação/afastamento com polinômio de tangente constante	*PGs/*	m	●	●	●	●
L	Número da subrotina	*PGAs/* Chamada de subrotina sem transferência de parâmetros (Página 182)	b	●	●	●	●
LEAD	Ângulo de avanço 1. Orientação da ferramenta 2. Polinômios de orientação	*PGAs/* Programação da orientação da ferramenta (A..., B..., C..., LEAD, TILT) (Página 329)	m	● -	● -	● -	● -
LEADOF	Acoplamento de valor mestre OFF	*PGAs/* Acoplamento axial de valor mestre (LEADON, LEADOF) (Página 532)		-	-	-	-
LEADON	Acoplamento de valor mestre ON	*PGAs/* Acoplamento axial de valor mestre (LEADON, LEADOF) (Página 532)		-	-	-	-
LENTOAX	Fornece informações sobre a associação dos comprimentos de ferramenta L1, L2 e L3 da ferramenta com a abscissa, ordenada e aplicada	*FB1(W1)*		●	●	●	●
LFOF [4]	Retrocesso rápido para rosqueamento OFF	*PGs/*	m	●	●	●	●
LFON	Retrocesso rápido para rosqueamento ON	*PGs/*	m	●	●	●	●
LFPOS	Retrocesso do eixo identificado com POLFMASK ou POLFMLIN na posição de eixo absoluta programada com POLF	*PGs/*	m	●	●	●	●
LFTXT	O plano do movimento de retrocesso na retração rápida é determinado a partir da tangente da trajetória e do atual sentido de ferramenta	*PGs/*	m	●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
LFWP	O plano do movimento de retrocesso na retração rápida é determinado através do atual plano de trabalho (G17/G18/G19)	*PGs/*	m	•	•	•	•
LIFTFAST	Retração rápida	*PGs/* Retração rápida do contorno (SETINT LIFTFAST, ALF) (Página 118)		•	•	•	•
LIMS	Limite de rotação com G96/G961 e G97	*PGs/*	m	•	•	•	•
LLI	Valor limite inferior de variáveis	*PGAs/* Atributo: Valores de limite (LLI, ULI) (Página 37)		•	•	•	•
LN	Logaritmo natural	*PGAs/* Funções de cálculo (Página 64)		•	•	•	•
LOCK	Bloqueio de ação síncrona com MD (parar ciclo de tecnologia)	*PGAs/* Bloquear, liberar, resetar (LOCK, UNLOCK, RESET) (Página 646)		•	•	•	•
LONGHOLE	Ciclo de modelo de fresamento de oblongos em uma circunferência	*BHDs//BHFs/*		-	-	-	-
LOOP	Introdução de um loop sem fim	*PGAs/* Loop de programa infinito (LOOP, ENDLOOP) (Página 101)		•	•	•	•

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
M0	Parada programada	*PGs/*		•	•	•	•
M1	Parada opcional	*PGs/*		•	•	•	•
M2	Fim do programa principal com retorno ao início do programa	*PGs/*		•	•	•	•
M3	Sentido de giro do fuso à direita (horário)	*PGs/*		•	•	•	•
M4	Sentido de giro do fuso à esquerda (anti-horário)	*PGs/*		•	•	•	•
M5	Parada do fuso	*PGs/*		•	•	•	•

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
M6	Troca de ferramentas	*PGs*/		•	•	•	•
M17	Fim de subrotina	*PGs*/		•	•	•	•
M19	Posicionamento de fuso na posição registrada no SD43240	*PGs*/		•	•	•	•
M30	Fim de programa, como o M2	*PGs*/		•	•	•	•
M40	Mudança automática da gama de velocidade	*PGs*/		•	•	•	•
M41 ... M45	Gama de velocidade 1 ... 5	*PGs*/		•	•	•	•
M70	Passagem para o modo de eixo	*PGs*/		•	•	•	•
MASLDEF	Definição de grupo de eixos mestres/escravos	*PGAs*/ Grupo mestre/escravo (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS) (Página 558)		•	•	•	•
MASLDEL	Separação de grupo de eixos mestres/escravos e cancelamento da definição do grupo	*PGAs*/ Grupo mestre/escravo (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS) (Página 558)		•	•	•	•
MASLOF	Desativação de um acoplamento temporário	*PGAs*/ Grupo mestre/escravo (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS) (Página 558)		•	•	•	•
MASLOFS	Desativação de um acoplamento temporário com parada automática do eixo escravo	*PGAs*/ Grupo mestre/escravo (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS) (Página 558)		•	•	•	•
MASLON	Ativação de um acoplamento temporário	*PGAs*/ Grupo mestre/escravo (MASLDEF, MASLDEL, MASLON, MASLOF, MASLOFS) (Página 558)		•	•	•	•
MATCH	Localização de uma String em Strings	*PGAs*/ Localizar caractere/String na String (INDEX, RINDEX, MINDEX, MATCH) (Página 81)		•	•	•	•

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
MAXVAL	Maior valor de duas variáveis (função aritm.)	*PGAs*/ Mínimo, máximo e área de variáveis (MINVAL, MAXVAL, BOUND) (Página 71)		●	●	●	●
MCALL	Chamada de subrotina modal	*PGAs*/ Chamada modal de subrotina (MCALL) (Página 189)		●	●	●	●
MEAC	Medição constante sem anulação de curso restante	*PGAs*/ Função de medição ampliada (MEASA, MEAWA, MEAC) (opcional) (Página 266)	b	-	-	-	-
MEAFRAME	Cálculo de Frame a partir de pontos de medição	*PGAs*/ Cálculo de Frame a partir de 3 pontos de medição no espaço (MEAFRAME) (Página 303)		●	●	●	●
MEAS	Medição com apalpador comutável	*PGAs*/ Medição com apalpador comutável (MEAS, MEAW) (Página 263)	b	●	●	●	●
MEASA	Medição com anulação de curso restante	*PGAs*/ Função de medição ampliada (MEASA, MEAWA, MEAC) (opcional) (Página 266)	b	-	-	-	-
MEASURE	Método de cálculo para medição de peça e medição de ferramenta	*FB2(M5)* Medição com apalpador comutável (MEAS, MEAW) (Página 263)		●	●	●	●
MEAW	Medição com apalpador comutável sem anulação de curso restante	*PGAs*/ Medição com apalpador comutável (MEAS, MEAW) (Página 263)	b	●	●	●	●
MEAWA	Medição sem anulação de curso restante	*PGAs*/ Função de medição ampliada (MEASA, MEAWA, MEAC) (opcional) (Página 266)	b	-	-	-	-
MI	Acesso aos dados de Frame: Espelhamento	*PGAs*/ Ler e alterar componentes de Frame (TR, FI, RT, SC, MI) (Página 294)		●	●	●	●
MINDEX	Determinação do índice de um caractere na String de entrada	*PGAs*/ Localizar caractere/String na String (INDEX, RINDEX, MINDEX, MATCH) (Página 81)		●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
MINVAL	Menor valor de duas variáveis (função aritm.)	*PGAs*/ Mínimo, máximo e área de variáveis (MINVAL, MAXVAL, BOUND) (Página 71)		●	●	●	●
MIRROR	Espelhamento programável	*PGAs*/	b	●	●	●	●
MMC	Chamada da janela de diálogo interativa na HMI a partir do programa de peça	*PGAs*/ Chamada interativa a partir do programa de peça (MMC) (Página 703)		●	●	●	●
MOD	Divisão Modulo	*PGAs*/ Funções de cálculo (Página 64)		●	●	●	●
MODAXVAL	Determinação da posição Modulo de um eixo rotativo Modulo	*PGAs*/ Funções de eixo (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) (Página 681)		●	●	●	●
MOV	Partida de eixo de posicionamento	*PGAs*/ Iniciar/parar eixo (MOV) (Página 614)		●	●	●	●
MSG	Mensagens programáveis	*PGs*/	m	●	●	●	●
MVTOOL	Comando de linguagem para movimentar uma ferramenta	*FBW*		●	●	●	●
N	Número de bloco secundário NC	*PGs*/		●	●	●	●
NCK	Especificação da área de validade de dados	*PGAs*/ Definição de variáveis de usuário (DEF) (Página 24)		●	●	●	●
NEWCONF	Aceitação dos dados de máquina modificados (corresponde à "Ativação do dado de máquina")	*PGAs*/ Ativar dados de máquina (NEWCONF) (Página 131)		●	●	●	●
NEWT	Criação de nova ferramenta	*PGAs*/ Ler e alterar componentes de Frame (TR, FI, RT, SC, MI) (Página 294)		●	●	●	●
NORM [4]	Ajuste normal dos pontos inicial e final com compensação de ferramenta	*PGs*/	m	●	●	●	●
NOT	NÃO lógico (Negation)	*PGAs*/ Operadores de comparação e operadores lógicos (Página 67)		●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
NPROT	Área de proteção específica de máquina ON/OFF	*PGAs*/ Ativar/desativar áreas de proteção (CPROT, NPROT) (Página 223)		•	•	•	•
NPROTDEF	Definição de uma área de proteção específica de máquina	*PGAs*/ Definição das áreas de proteção (CPROTDEF, NPROTDEF) (Página 219)		•	•	•	•
NUMBER	Converte a String de entrada em número	*PGAs*/ Conversão de tipos de STRING (NUMBER, ISNUMBER, AXNAME) (Página 77)		•	•	•	•
OEMIPO1	Interpolação OEM 1	*PGAs*/ Funções especiais para usuários OEM (OEMIPO1, OEMIPO2, G810 até G829) (Página 276)	m	•	•	•	•
OEMIPO2	Interpolação OEM 2	*PGAs*/ Funções especiais para usuários OEM (OEMIPO1, OEMIPO2, G810 até G829) (Página 276)	m	•	•	•	•
OF	Palavra-chave na bifurcação CASE	*PGAs*/ Bifurcação do programa (CASE ... OF ... DEFAULT ...) (Página 89)		•	•	•	•
OFFN	Sobremetal para contorno programado	*PGs*/	m	•	•	•	•
OMA1	Endereço OEM 1		m	•	•	•	•
OMA2	Endereço OEM 2		m	•	•	•	•
OMA3	Endereço OEM 3		m	•	•	•	•
OMA4	Endereço OEM 4		m	•	•	•	•
OMA5	Endereço OEM 5		m	•	•	•	•
OR	Operador lógico, operador lógico OU	*PGAs*/ Operadores de comparação e operadores lógicos (Página 67)		•	•	•	•
ORIAXES	Interpolação linear dos eixos de máquina ou eixos de orientação	*PGAs*/ Programação dos eixos de orientação (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2) (Página 339)	m	•	•	•	•

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
ORIAXPOS	Ângulo de orientação através de eixos virtuais de orientação com posições de eixo rotativo		m	•	•	•	•
ORIC [4]	As mudanças de orientação em cantos externos são sobrepostas no bloco circular a ser inserido	*PGAsl* Orientação da ferramenta (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST) (Página 434)	m	•	•	•	•
ORICONCCW	Interpolação em uma superfície periférica circular em sentido anti-horário	*PGAsl/FB3(F3)* Programação da orientação ao longo de uma superfície periférica cônica (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO) (Página 342)	m	•	•	•	•
ORICONCW	Interpolação em uma superfície periférica circular em sentido horário	*PGAsl/FB3(F4)* Programação da orientação ao longo de uma superfície periférica cônica (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO) (Página 342)	m	•	•	•	•
ORICONIO	Interpolação em uma superfície periférica circular com especificação de uma orientação intermediária	*PGAsl/FB3(F4)* Programação da orientação ao longo de uma superfície periférica cônica (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO) (Página 342)	m	•	•	•	•
ORICONTO	Interpolação em uma superfície periférica circular na transição tangencial (especificação da orientação final)	*PGAsl/FB3(F5)* Programação da orientação ao longo de uma superfície periférica cônica (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO) (Página 342)	m	•	•	•	•
ORICURVE	Interpolação da orientação com especificação do movimento de dois pontos de contato da ferramenta	*PGAsl/FB3(F6)* Especificação de orientação de dois pontos de contato (ORICURVE, PO[XH]=, PO[YH]=, PO[ZH]=) (Página 346)	m	•	•	•	•
ORID	As mudanças de orientação são executadas antes de um bloco circular	*PGAsl* Orientação da ferramenta (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST) (Página 434)	m	•	•	•	•

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
ORIEULER	Ângulo de orientação através de ângulo euleriano	*PGAs*/ Programação dos eixos de orientação (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2) (Página 339)	m	•	•	•	•
ORIMKS	Orientação de ferramenta no sistema de coordenadas da máquina	*PGAs*/ Relação dos eixos de orientação (ORIWKS, ORIMKS) (Página 337)	m	•	•	•	•
ORIPATH	Orientação da ferramenta relativa à trajetória	*PGAs*/ Rotação da orientação da ferramenta relativa à trajetória (ORIPATH, ORIPATHS, ângulo de rotação) (Página 355)	m	•	•	•	•
ORIPATHS	Orientação de ferramenta relativa à trajetória, uma dobra é suavizada no decurso da orientação	*PGAs*/ Rotação da orientação da ferramenta relativa à trajetória (ORIPATH, ORIPATHS, ângulo de rotação) (Página 355)	m	•	•	•	•
ORIPLANE	Interpolação em um plano (corresponde ao ORIVECT) Interpolação de grande circunferência	*PGAs*/ Programação da orientação ao longo de uma superfície periférica cônica (ORIPLANE, ORICONCW, ORICONCCW, ORICONTO, ORICONIO) (Página 342)	m	•	•	•	•
ORIRESET	Posição inicial da orientação de ferramenta com até 3 eixos de orientação	*PGAs*/ Variantes da programação da orientação e posição básica (ORIRESET) (Página 327)		•	•	•	•
ORIROTA	Ângulo de rotação de um sentido de rotação especificado como absoluto	*PGAs*/ Rotações da orientação da ferramenta (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) (Página 350)	m	•	•	•	•
ORIROTC	Vetor de rotação tangencial à tangente da trajetória	*PGAs*/ Rotações da orientação da ferramenta (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) (Página 350)	m	•	•	•	•

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
ORIROTR	Ângulo de rotação relativo ao plano entre a orientação inicial e a orientação final	*PGAs*/ Rotações da orientação da ferramenta (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) (Página 350)	m	•	•	•	•
ORIROTT	Ângulo de rotação relativo à alteração do vetor de orientação	*PGAs*/ Rotações da orientação da ferramenta (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) (Página 350)	m	•	•	•	•
ORIRPY	Ângulo de orientação através de ângulo RPY (XYZ)	*PGAs*/ Programação dos eixos de orientação (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2) (Página 339)	m	•	•	•	•
ORIRPY2	Ângulo de orientação através de ângulo RPY (ZYX)	*PGAs*/ Programação dos eixos de orientação (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2) (Página 339)	m	•	•	•	•
ORIS	Alteração de orientação	*PGAs*/ Orientação da ferramenta (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST) (Página 434)	m	•	•	•	•
ORISOF [4]	Suavização do decurso de orientação OFF	*PGAs*/ Suavização do decurso de orientação (ORISON, ORISOF) (Página 365)	m	•	•	•	•
ORISON	Suavização do decurso de orientação ON	*PGAs*/ Suavização do decurso de orientação (ORISON, ORISOF) (Página 365)	m	•	•	•	•
ORIVECT	Interpolação de grande circunferência (idêntico ao ORIPLANE)	*PGAs*/ Programação dos eixos de orientação (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2) (Página 339)	m	•	•	•	•

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
ORIVIRT1	Ângulo de orientação através de eixos virtuais de orientação (Definition 1)	*PGAs*/ Programação dos eixos de orientação (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2) (Página 339)	m	●	●	●	●
ORIVIRT2	Ângulo de orientação através de eixos virtuais de orientação (Definition 1)	*PGAs*/ Programação dos eixos de orientação (ORIAXES, ORIVECT, ORIEULER, ORIRPY, ORIRPY2, ORIVIRT1, ORIVIRT2) (Página 339)	m	●	●	●	●
ORIWKS [4]	Orientação de ferramenta no sistema de coordenadas da peça de trabalho	*PGAs*/ Relação dos eixos de orientação (ORIWKS, ORIMKS) (Página 337)	m	●	●	●	●
OS	Oscilação ativada/desativada	*PGAs*/ Oscilação assíncrona (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB) (Página 653)		-	-	-	-
OSB	Oscilação: Ponto de partida	*FB2(P5)*	m	-	-	-	-
OSC	Suavização constante da orientação da ferramenta	*PGAs*/ Orientação da ferramenta (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST) (Página 434)	m	●	●	●	●
OSCILL	Axis: 1 - 3 eixos de penetração	*PGAs*/ Oscilação controlada através de ações síncronas (OSCILL) (Página 659)	m	-	-	-	-
OSCTRL	Opções da oscilação	*PGAs*/ Oscilação assíncrona (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB) (Página 653)	m	-	-	-	-
OSD	Suavização da orientação de ferramenta através da especificação da extensão de suavização com SD.	*PGAs*/ Orientação da ferramenta (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST) (Página 434)	m	●	●	●	●
OSE	Ponto final da oscilação	*PGAs*/ Oscilação assíncrona (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB) (Página 653)	m	-	-	-	-

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
OSNSC	Oscilação: Número de passadas finais	*PGAs*/ Oscilação assíncrona (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB) (Página 653)	m	-	-	-	-
OSOF [4]	Suavização da orientação de ferramenta OFF	*PGAs*/ Oscilação assíncrona (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB) (Página 653)	m	●	●	●	●
OSP1	Oscilação: ponto de reversão esquerdo	*PGAs*/ Oscilação assíncrona (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB) (Página 653)	m	-	-	-	-
OSP2	Ponto de reversão direito da oscilação	*PGAs*/ Oscilação assíncrona (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB) (Página 653)	m	-	-	-	-
OSS	Suavização da orientação da ferramenta no fim do bloco	*PGAs*/ Orientação da ferramenta (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST) (Página 434)	m	●	●	●	●
OSSE	Suavização da orientação de ferramenta no início e no fim do bloco	*PGAs*/ Orientação da ferramenta (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST) (Página 434)	m	●	●	●	●
OST	Suavização da orientação de ferramenta através da especificação da tolerância angular em graus com SD (desvio máximo do decurso de orientação programado)	*PGAs*/ Orientação da ferramenta (ORIC, ORID, OSOF, OSC, OSS, OSSE, ORIS, OSD, OST) (Página 434)	m	●	●	●	●
OST1	Oscilação: Ponto de parada no ponto de reversão esquerdo	*PGAs*/ Oscilação assíncrona (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB) (Página 653)	m	-	-	-	-
OST2	Oscilação: Ponto de parada no ponto de reversão direito	*PGAs*/ Oscilação assíncrona (OS, OSP1, OSP2, OST1, OST2, OSCTRL, OSNSC, OSE, OSB) (Página 653)	m	-	-	-	-

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
OTOL	Tolerância de orientação para funções de compressor, suavização de orientação e tipos de suavização	*PGAs*/ Tolerância de contorno/orientação programável (CTOL, OTOL, ATOL) (Página 503)		-	●	-	●
OVR	Correção de rotação	*PGAs*/	m	●	●	●	●
OVRA	Correção de rotação axial	*PGAs*/	m	●	●	●	●
OVRRAP	Correção do avanço rápido	*PGAs*/	m	●	●	●	●
P	Número de processamentos da subrotina	*PGAs*/ Número de repetições de programa (P) (Página 187)		●	●	●	●
PAROT	Alinhamento do sistema de coordenadas à peça de trabalho	*PGs*/	m	●	●	●	●
PAROTOF	Desativação da rotação de Frame relativa à peça de trabalho	*PGs*/	m	●	●	●	●
PCALL	Chamada de subrotinas com indicação absoluta do caminho e transferência de parâmetros	*PGAs*/ Chamada de subrotina com indicação de caminho e parâmetros (PCALL) (Página 195)		●	●	●	●
PDELAYOF	Retardamento na estampagem OFF	*PGAs*/ Estampagem e puncionamento ativado ou desativado (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC) (Página 667)	m	-	-	-	-
PDELAYON [4]	Retardamento na estampagem ON	*PGAs*/ Estampagem e puncionamento ativado ou desativado (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC) (Página 667)	m	-	-	-	-
PHU	Unidade física de uma variável	*PGAs*/ Definição de variáveis de usuário (DEF) (Página 24)		●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
PL	1. B-Spline: Distância entre os nós 2. Interpolação de polinômios: Comprimento do intervalo de parâmetros na interpolação de polinômios	*PGAs/* 1. Interpolação de Spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) (Página 237) 2. Interpolação de polinômios (POLY, POLYPATH, PO, PL) (Página 254)	b	- -	○ -	- -	○ -
PM	por minuto	*PGs/*		●	●	●	●
PO	Coeficiente de polinômio na interpolação de polinômios	*PGAs/* Interpolação de polinômios (POLY, POLYPATH, PO, PL) (Página 254)	b	-	-	-	-
POCKET3	Ciclo de fresamento, bolsão retangular (qualquer fresa)	*BHDs//BHFs/*		●	●	●	●
POCKET4	Ciclo de fresamento, bolsão circular (qualquer fresa)	*BHDs//BHFs/*		●	●	●	●
POLF	Posição de retrocesso LIFTFAST	*PGs//PGAs/*	m	●	●	●	●
POLFA	Início da posição de retrocesso dos eixos individuais com $AA_ESR_TRIGGER	*PGs/*	m	●	●	●	●
POLFMASK	Liberar eixos para o retrocesso sem haver relação entre eixos	*PGs/*	m	●	●	●	●
POLFMLIN	Liberação de eixos para o retrocesso com relação linear entre os eixos	*PGs/*	m	●	●	●	●
POLY	Interpolação de polinômios	*PGAs/* Interpolação de polinômios (POLY, POLYPATH, PO, PL) (Página 254)	m	-	-	-	-
POLYPATH	A interpolação de polinômios pode ser selecionada para os grupos de eixos AXIS ou VECT	*PGAs/* Interpolação de polinômios (POLY, POLYPATH, PO, PL) (Página 254)	m	-	-	-	-

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
PON	Estampagem ON	*PGAs*/ Estampagem e puncionamento ativado ou desativado (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC) (Página 667)	m	-	-	-	-
PONS	Estampagem ON no ciclo IPO	*PGAs*/ Estampagem e puncionamento ativado ou desativado (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC) (Página 667)	m	-	-	-	-
POS	Posicionamento de eixo	*PGs*/		●	●	●	●
POSA	Posicionamento de eixo além dos limites de bloco	*PGs*/		●	●	●	●
POSM	Posicionamento do magazine	*FBW*		●	●	●	●
POSP	Posicionamento em segmentos (oscilação)	*PGs*/		●	●	●	●
POSRANGE	Determinação se a atual posição nominal interpolada de um eixo encontra-se em uma janela e uma posição de referência pré-definida	*PGAs*/ Posição na área de referência especificada (POSRANGE) (Página 613)		●	●	●	●
POT	Quadrado (função aritmética)	*PGAs*/ Funções de cálculo (Página 64)		●	●	●	●
PR	Por rotação	*PGs*/		●	●	●	●
PREPRO	Identificação de subrotinas com preparação	*PGAs*/ Identificar subrotinas com preparação (PREPRO) (Página 172)		●	●	●	●
PRESETON	Definição de valores reais para eixos programados	*PGAs*/ Deslocamento Preset (PRESETON) (Página 301)		●	●	●	●
PRIO	Palavra-chave para definir a prioridade no tratamento de interrupções	*PGAs*/ Atribuição e partida de rotinas de interrupção (SETINT, PRIO, BLSYNC) (Página 114)		●	●	●	●
PROC	Primeira instrução de um programa	*PGAs*/ Chamada de subrotina com indicação de caminho e parâmetros (PCALL) (Página 195)		●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
PTP	Movimento ponto a ponto	*PGAs*/ Deslocamento PTP cartesiano (Página 385)	m	●	●	●	●
PTPG0	Movimento ponto a ponto somente com G0, senão CP	*PGAs*/ PTP com TRANSMIT (Página 390)	m	●	●	●	●
PUNCHACC	Aceleração em função do curso para puncionamento	*PGAs*/ Estampagem e puncionamento ativado ou desativado (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC) (Página 667)		-	-	-	-
PUTFTOC	Correção fina de ferramenta para dressagem paralela	*PGAs*/ Correção de ferramenta Online (PUTFTOCF, FCTDEF, PUTFTOC, FTOCON, FTOCOF) (Página 415)		●	●	●	●
PUTFTOCF	Correção fina de ferramenta em função de uma função definida com FCTDEF para dressagem paralela	*PGAs*/ Correção de ferramenta Online (PUTFTOCF, FCTDEF, PUTFTOC, FTOCON, FTOCOF) (Página 415)		●	●	●	●
PW	B-Spline, peso do ponto	*PGAs*/ Interpolação de Spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) (Página 237)	b	-	○	-	○
QECLRNOF	Aprendizado da compensação de erro de quadrante OFF	*PGAs*/ Adaptação de curvas características de compensação (QECLRNON, QECLRNOF) (Página 701)		●	●	●	●
QECLRNON	Aprendizado da compensação de erro de quadrante ON	*PGAs*/ Adaptação de curvas características de compensação (QECLRNON, QECLRNOF) (Página 701)		●	●	●	●
QU	Emissão rápida de função (auxiliar) adicional	*PGs*/		●	●	●	●
R...	Parâmetro de cálculo também como identificador de eixo ajustável e com extensão numérica	*PGAs*/ Variáveis de usuário pré-definidas: Parâmetros de cálculo (R) (Página 20)		●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
RAC	Programação de raio específica de eixo, absoluta e por blocos	*PGs/*	b	•	•	•	•
RDISABLE	Bloqueio de leitura (entrada)	*PGAs/* Definir bloqueio de leitura (RDISABLE) (Página 592)		•	•	•	•
READ	Lê uma ou várias linhas de um arquivo especificado e carrega estas informações no campo	*PGAs/* Ler linhas no arquivo (READ) (Página 137)		•	•	•	•
REAL	Tipo de dado: Variável de vírgula flutuante com sinal (números reais)	*PGAs/* Definição de variáveis de usuário (DEF) (Página 24)		•	•	•	•
REDEF	Ajuste para dados de máquina, elementos de linguagem NC e variáveis de sistema que são exibidos para determinados grupos de usuários	*PGAs/* Redefinição de variáveis de sistema, variáveis de usuário e comandos de linguagem NC (REDEF) (Página 30)		•	•	•	•
RELEASE	Liberação de eixos de máquina para troca de eixos	*PGAs/* Troca de eixos, troca de fusos (RELEASE, GET, GETD) (Página 124)		•	•	•	•
REP	Palavra-chave para inicialização de todos elementos de um campo com o mesmo valor	*PGAs/* Definição e inicialização de variáveis de campo (DEF, SET, REP) (Página 47)		•	•	•	•
REPEAT	Repetição de um loop de programa	*PGAs/* Repetição de partes do programa (REPEAT, REPEATB, ENDLABEL, P) (Página 91)		•	•	•	•
REPEATB	Repetição de uma linha do programa	*PGAs/* Repetição de partes do programa (REPEAT, REPEATB, ENDLABEL, P) (Página 91)		•	•	•	•
REPOSA	Reaproximação até o contorno linear com todos os eixos	*PGAs/* Reaproximação no contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN) (Página 488)	b	•	•	•	•

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
REPOSH	Reaproximação até o contorno com semicírculo	*PGAs*/ Reaproximação no contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN) (Página 488)	b	●	●	●	●
REPOSHA	Reaproximação até o contorno com todos os eixos; eixos geométricos em semicírculo	*PGAs*/ Reaproximação no contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN) (Página 488)	b	●	●	●	●
REPOSL	Reaproximação até o contorno linear	*PGAs*/ Reaproximação no contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN) (Página 488)	b	●	●	●	●
REPOSQ	Reaproximação até o contorno em quadrante	*PGAs*/ Reaproximação no contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN) (Página 488)	b	●	●	●	●
REPOSQA	Reaproximação até o contorno linear com todos os eixos; eixos geométricos em quadrante	*PGAs*/ Reaproximação no contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN) (Página 488)	b	●	●	●	●
RESET	Resetamento de ciclo de tecnologia	*PGAs*/ Bloquear, liberar, resetar (LOCK, UNLOCK, RESET) (Página 646)		●	●	●	●
RESETMON	Comando de linguagem para ativação de valor nominal	*FBW*		●	●	●	●
RET	Fim de subrotina	*PGAs*/ Salto de retorno parametrizável da subrotina (RET ...) (Página 175)		●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
RIC	Programação em raio relativa por bloco e específica de eixo	*PGs*/ Troca de eixos, troca de fusos (RELEASE, GET, GETD) (Página 124)	b	•	•	•	•
RINDEX	Determinação do índice de um caractere na String de entrada	*PGAs*/ Localizar caractere/String na String (INDEX, RINDEX, MINDEX, MATCH) (Página 81)		•	•	•	•
RMB	Reaproximação no ponto inicial do bloco	*PGAs*/ Reaproximação no contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN) (Página 488)	m	•	•	•	•
RME	Reaproximação no ponto final do bloco	*PGAs*/ Reaproximação no contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN) (Página 488)	m	•	•	•	•
RMI [4]	Reaproximação no ponto de interrupção	*PGAs*/ Reaproximação no contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN) (Página 488)	m	•	•	•	•
RMN	Reaproximação no ponto de percurso mais próximo	*PGAs*/ Reaproximação no contorno (REPOSA, REPOSL, REPOSQ, REPOSQA, REPOSH, REPOSHA, DISR, DISPR, RMI, RMB, RME, RMN) (Página 488)	m	•	•	•	•
RND	Arredondamento do canto do contorno	*PGs*/	b	•	•	•	•
RNDM	Arredondamento modal	*PGs*/	m	•	•	•	•
ROT	Rotação programável	*PGs*/	b	•	•	•	•
ROTS	Rotações de Frame programáveis com ângulos espaciais	*PGs*/	b	•	•	•	•
ROUND	Arredondamento das casas decimais	*PGAs*/ Funções de cálculo (Página 64)		•	•	•	•

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
ROUNDUP	Arredondamento para cima de um valor de entrada	*PGAs*/ Arredondamento (ROUNDUP) (Página 148)		●	●	●	●
RP	Raio polar	*PGs*/	m/b	●	●	●	●
RPL	Rotação no plano	*PGs*/	b	●	●	●	●
RT	Parâmetro para acesso aos dados de Frame: Rotação	*PGAs*/ Ler e alterar componentes de Frame (TR, FI, RT, SC, MI) (Página 294)		●	●	●	●
RTLIOF	G0 sem interpolação linear (interpolação de eixos individuais)	*PGs*/	m	●	●	●	●
RTLION	G0 com interpolação linear	*PGs*/	m	●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
S	Rotação do fuso (com G4, H96/G961 tem outro significado)	*PGs*/	m/b	●	●	●	●
SAVE	Atributo para salvar informações em chamadas de subrotinas	*PGAs*/ Salvar funções G modais (SAVE) (Página 161)		●	●	●	●
SBLOF	Supressão de bloco a bloco	*PGAs*/ Supressão de processamento bloco a bloco (SBLOF, SBLON) (Página 162)		●	●	●	●
SBLON	Cancelamento da supressão de bloco a bloco	*PGAs*/ Supressão de processamento bloco a bloco (SBLOF, SBLON) (Página 162)		●	●	●	●
SC	Parâmetro para acesso aos dados de Frame: Escala	*PGAs*/ Ler e alterar componentes de Frame (TR, FI, RT, SC, MI) (Página 294)		●	●	●	●
SCALE	Escala programável	*PGs*/	b	●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
SCC	Atribuição seletiva de um eixo transversal ao G96/G961/G962. Os identificadores de eixo podem ser de eixo geométrico, de canal ou de máquina.	*PGsl*		●	●	●	●
SCPARA	Programação de bloco de parâmetros servo	*PGAsl* Bloco programável de parâmetros servo (SCPARA) (Página 282)		●	●	●	●
SD	Grau de Spline	*PGAsl* Interpolação de Spline (ASPLINE, BSPLINE, CSPLINE, BAUTO, BNAT, BTAN, EAUTO, ENAT, ETAN, PW, SD, PL) (Página 237)	b	-	○	-	○
SEFORM	Instrução de estruturação no editor Step, para gerar a exibição de passos para HMI-Advanced	*PGAsl* Instrução de estruturação no editor Step (SEFORM) (Página 217)		●	●	●	●
SET	Palavra-chave para inicialização de todos elementos de um campo com valores listados	*PGAsl* Definição e inicialização de variáveis de campo (DEF, SET, REP) (Página 47)		●	●	●	●
SETAL	Definição de alarme	*PGAsl* Alarmes (SETAL) (Página 711)		●	●	●	●
SETDNO	Atribuição de número D do corte (CE) de uma ferramenta (T)	*PGAsl* Atribuição livre de números D: Renomear números D (GETDNO, SETDNO) (Página 442)		●	●	●	●
SETINT	Definição de qual rotina de interrupção deverá ser ativada quando existir uma entrada NCK	*PGAsl* Atribuição e partida de rotinas de interrupção (SETINT, PRIO, BLSYNC) (Página 114)		●	●	●	●
SETM	Definição de marcadores em canal próprio	PGAsl Coordenação de programa (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM) (Página 106)		-	-	-	-
SETMS	Retorna para o fuso mestre definido no dado de máquina			●	●	●	●
SETMS(n)	O fuso n deve valer como fuso mestre	*PGsl*		●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
SETMTH	Definição de número de porta-ferramenta mestre	*FBW*		●	●	●	●
SETPIECE	Consideração do número de peças para todas ferramentas atribuídas ao fuso	*FBW*		●	●	●	●
SETTA	Definição de uma ferramenta do grupo de desgaste como ativa	*FBW*		●	●	●	●
SETTCOR	Alteração de componentes de ferramenta sob consideração de todas condições gerais	*FB1(W1)*		●	●	●	●
SETTIA	Definição de uma ferramenta do grupo de desgaste como inativa	*FBW*		●	●	●	●
SF	Deslocamento do ponto de partida para rosqueamento	*PGsl*	m	●	●	●	●
SIN	Seno (função trigonométrica)	*PGAsl* Funções de cálculo (Página 64)		●	●	●	●
SIRELAY	Ativação das funções de segurança parametrizadas com SIRELIN, SIRELOUT e SIRELTIME	*FBSIsl*		-	-	-	-
SIRELIN	Inicialização de grandezas de entrada do módulo de função	*FBSIsl*		-	-	-	-
SIRELOUT	Inicialização de grandezas de saída do módulo de função	*FBSIsl*		-	-	-	-
SIRELTIME	Inicialização do Timer do módulo de função	*FBSIsl*		-	-	-	-
SLOT1	Ciclo de modelo de fresamento, ranhuras em uma circunferência	*BHDsl/BHFsl*		●	●	●	●
SLOT2	Ciclo de modelo de fresamento, ranhura circular	*BHDsl/BHFsl*		●	●	●	●
SOFT	Aceleração de trajetória suave	*PGsl*	m	●	●	●	●
SOFTA	Ativação da aceleração de eixo brusca para os eixos programados	*PGsl*		●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
SON	Puncionamento ON	*PGAs/* Estampagem e puncionamento ativado ou desativado (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC) (Página 667)	m	-	-	-	-
SONS	Puncionamento ON no ciclo IPO	*PGAs/* Estampagem e puncionamento ativado ou desativado (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC) (Página 667)	m	-	-	-	-
SPATH [4]	A referência de percurso para eixos FGROUP é o comprimento do arco	*PGAs/* Referência ajustável do percurso (SPATH, UPATH) (Página 260)	m	●	●	●	●
SPCOF	Comutação do fuso mestre ou fuso (n) de controle de posição para controle de rotação	*PGs/*	m	●	●	●	●
SPCON	Comutação do fuso mestre ou fuso (n) de controle de rotação para controle de posição	*PGAs/*	m	●	●	●	●
SPI	Converte o número de fuso em identificador de eixo	*PGAs/* Funções de eixo (AXNAME, AX, SPI, AXTOSPI, ISAXIS, AXSTRING, MODAXVAL) (Página 681)		●	●	●	●
SPIF1 [4]	Entradas/saídas NCK rápidas para estampagem/puncionamento Byte 1	*FB2(N4)*	m	-	-	-	-
SPIF2	Entradas/saídas NCK rápidas para estampagem/puncionamento Byte 2	*FB2(N4)*	m	-	-	-	-
SPLINEPATH	Definição de grupo de Spline	*PGAs/* Agrupamento de Spline (SPLINEPATH) (Página 249)		-	○	-	○
SPN	Número de trechos por bloco	*PGAs/* Preparação automática do curso (Página 672)	b	-	-	-	-

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
SPOF [4]	Curso OFF, estampagem, puncionamento OFF	*PGAs*/ Estampagem e puncionamento ativado ou desativado (SPOF, SON, PON, SONS, PONS, PDELAYON, PDELAYOF, PUNCHACC) (Página 667)	m	-	-	-	-
SPOS	Posição do fuso	*PGs*/	m	●	●	●	●
SPOSA	Posição do fuso além dos limites do bloco	*PGs*/	m	●	●	●	●
SPP	Comprimento de um trecho	*PGAs*/ Preparação automática do curso (Página 672)	m	-	-	-	-
SQRT	Raiz quadrada (função aritmética) (square root)	*PGAs*/ Funções de cálculo (Página 64)		●	●	●	●
SR	Curso de retrocesso oscilante para ação síncrona	*PGs*/	b	-	-	-	-
SRA	Curso de retrocesso oscilante na entrada externa axial para ação síncrona	*PGs*/	m	-	-	-	-
ST	Tempo de passada final oscilante para ação síncrona	*PGs*/	b	-	-	-	-
STA	Tempo de passada final oscilante axial para ação síncrona	*PGs*/	m	-	-	-	-
START	Inicialização dos programas selecionados em vários canais, simultaneamente a partir do programa em andamento	*PGAs*/ Coordenação de programa (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM) (Página 106)		-	-	-	-
STARTFIFO [4]	Execução; paralelo à isso, abastecimento da memória de pré-processamento	*PGAs*/ Execução do programa com memória de pré-processamento (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE) (Página 477)	m	●	●	●	●
STAT	Posição das articulações	*PGAs*/ Deslocamento PTP cartesiano (Página 385)	b	●	●	●	●
STOLF	Fator de tolerância G0	*PGAs*/ Tolerância em movimentos G0 (STOLF) (Página 507)	m	-	-	-	-

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
STOPFIFO	Parada do processamento; abastecimento da memória de pré-processamento até ser detectado o STARTFIFO, memória cheia ou fim de programa	*PGAs*/ Execução do programa com memória de pré-processamento (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE) (Página 477)	m	●	●	●	●
STOPRE	Parada de pré-processamento até todos os bloco preparados serem executados pelo processamento principal	*PGAs*/ Execução do programa com memória de pré-processamento (STOPFIFO, STARTFIFO, FIFOCTRL, STOPRE) (Página 477)		●	●	●	●
STOPREOF	Cancelamento da parada de pré-processamento	*PGAs*/ Cancelar a parada de pré-processamento (STOPREOF) (Página 593)		●	●	●	●
STRING	Tipo de dado: Sequência de caracteres	*PGAs*/ Definição de variáveis de usuário (DEF) (Página 24)		●	●	●	●
STRINGFELD	Seleção de um caractere individual a partir do campo de String programado	*PGAs*/ Seleção de um caractere individual (STRINGVAR, STRINGFELD) (Página 83)		●	●	●	●
STRINGIS	Verifica o escopo de linguagem NC disponível e especialmente verifica a existência, validade, definição e ativação dos nomes de ciclo NC, variáveis de usuário, macros e nomes de Label pertencentes a este comando.	*PGAs*/ Controlar o escopo disponível de linguagem NC (STRINGIS) (Página 695)		●	●	●	●
STRINGVAR	Seleção de um caractere individual a partir da String programada	*PGAs*/ Seleção de um caractere individual (STRINGVAR, STRINGFELD) (Página 83)		-	-	-	-
STRLEN	Determinação do comprimento de uma String	*PGAs*/ Definir o tamanho de uma Strings (STRLEN) (Página 81)		●	●	●	●
SUBSTR	Determinação do índice de um caractere na String de entrada	*PGAs*/ Seleção de uma String parcial (SUBSTR) (Página 83)		●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
SUPA	Supressão do atual deslocamento de ponto zero, inclusive os deslocamentos programados, Frames de sistema, deslocamentos com manivela eletrônica (DRF), deslocamento de ponto zero externo e movimento sobreposto	*PGs/*	b	●	●	●	●
SVC	Velocidade de corte da ferramenta	*PGs/*	m	●	●	●	●
SYNFCT	Avaliação de um polinômio em função de uma condição na ação síncrona de movimentos	*PGAs/* Função sincronizada (SYNFCT) (Página 599)		●	●	●	●
SYNR	A leitura da variável é síncrona, isto é, ocorre no momento da execução	*PGAs/* Definição de variáveis de usuário (DEF) (Página 24)		●	●	●	●
SYNRW	A leitura e gravação da variável são sincronizadas, isto é, ocorrem no momento da execução	*PGAs/* Definição de variáveis de usuário (DEF) (Página 24)		●	●	●	●
SYNW	A gravação da variável é sincronizada, isto é, ocorre no momento da execução	*PGAs/* Definição de variáveis de usuário (DEF) (Página 24)		●	●	●	●
T	Chamada de ferramenta (a troca somente ocorre se estiver definida no dado de máquina; senão será necessário o comando M6)	*PGs/*		●	●	●	●
TAN	Tangente (função trigonométrica)	*PGAs/* Funções de cálculo (Página 64)		●	●	●	●
TANG	Definição do grupo de eixos do acompanhamento tangencial	*PGAs/* Controle tangencial (TANG, TANGON, TANGOF, TLIFT, TANGDEL) (Página 465)		-	-	-	-
TANGDEL	Cancelamento da definição do grupo de eixos do acompanhamento tangencial	*PGAs/* Controle tangencial (TANG, TANGON, TANGOF, TLIFT, TANGDEL) (Página 465)		-	-	-	-

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
TANGOF	Acompanhamento tangencial OFF	*PGAsl* Controle tangencial (TANG, TANGON, TANGOF, TLIFT, TANGDEL) (Página 465)		-	-	-	-
TANGON	Acompanhamento tangencial ON	*PGAsl* Controle tangencial (TANG, TANGON, TANGOF, TLIFT, TANGDEL) (Página 465)		-	-	-	-
TCA	Seleção de ferramenta / troca de ferramentas independente do estado da ferramenta	*FBW*		●	●	●	●
TCARR	Solicitação de porta-ferramenta (número "m")	*PGAsl* Correção de comprimento de ferramenta para porta-ferramentas orientáveis (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ) (Página 451)		-	●	-	●
TCI	Troca a ferramenta do alojamento intermediário para o magazine	*FBW*		●	●	●	●
TCOABS [4]	Determinação de componentes de comprimento da ferramenta a partir da atual orientação de ferramenta	*PGAsl* Correção de comprimento de ferramenta para porta-ferramentas orientáveis (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ) (Página 451)	m	-	●	-	●
TCOFR	Determinação de componentes de comprimento da ferramenta a partir da orientação do Frame ativo	*PGAsl* Correção de comprimento de ferramenta para porta-ferramentas orientáveis (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ) (Página 451)	m	-	●	-	●
TCOFRX	Determinação da orientação de ferramenta de um Frame ativo na seleção de ferramenta, a ferramenta aponta para o sentido X	*PGAsl* Correção de comprimento de ferramenta para porta-ferramentas orientáveis (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ) (Página 451)	m	-	●	-	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
TCOFRY	Determinação da orientação de ferramenta de um Frame ativo na seleção de ferramenta, a ferramenta aponta para o sentido Y	*PGAs*/ Correção de comprimento de ferramenta para porta-ferramentas orientáveis (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ) (Página 451)	m	-	●	-	●
TCOFRZ	Determinação da orientação de ferramenta de um Frame ativo na seleção de ferramenta, a ferramenta aponta para o sentido Z	*PGAs*/ Correção de comprimento de ferramenta para porta-ferramentas orientáveis (TCARR, TCOABS, TCOFR, TCOFRX, TCOFRY, TCOFRZ) (Página 451)	m	-	●	-	●
THETA	Ângulo de giro	*PGAs*/ Rotações da orientação da ferramenta (ORIROTA, ORIROTR, ORIROTT, ORIROTC, THETA) (Página 350)	b	●	●	●	●
TILT	Ângulo lateral	*PGAs*/ Programação da orientação da ferramenta (A..., B..., C..., LEAD, TILT) (Página 329)	m	●	●	●	●
TLIFT	Inserção de bloco intermediário em cantos de contorno para controle tangencial	*PGAs*/ Controle tangencial (TANG, TANGON, TANGOF, TLIFT, TANGDEL) (Página 465)		-	-	-	-
TMOF	Cancelamento da monitoração de ferramentas	*PGAs*/ Monitoração de ferramenta específica de retificação no programa de peça (TMON, TMOF) (Página 679)		●	●	●	●
TMON	Ativação da monitoração de ferramentas	*PGAs*/ Monitoração de ferramenta específica de retificação no programa de peça (TMON, TMOF) (Página 679)		●	●	●	●
TO	Identifica o valor final em um loop de contagem FOR	*PGAs*/ Loop contador (FOR ... TO ..., ENDFOR) (Página 102)		●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
TOFF	Offset de comprimento de ferramenta no sentido do componente de comprimento da ferramenta, que atua paralelo ao eixo geométrico especificado no índice.	*PGs/*	m	•	•	•	•
TOFFL	Offset de comprimento de ferramenta no sentido do componente de comprimento da ferramenta L1, L2 ou L3	*PGs/*	m	•	•	•	•
TOFFOF	Resetamento da correção de comprimento de ferramenta Online	*PGAs/* Correção Online de comprimento de ferramenta (TOFFON, TOFFOF) (Página 455)		•	•	•	•
TOFFON	Ativação da correção de comprimento de ferramenta Online	*PGAs/* Correção Online de comprimento de ferramenta (TOFFON, TOFFOF) (Página 455)		•	•	•	•
TOFFR	Offset do raio da ferramenta	*PGs/*	m	•	•	•	•
TOFRAME	Alinhamento do eixo Z do WCS através da rotação de Frame paralelamente à orientação de ferramenta	*PGs/*	m	•	•	•	•
TOFRAMEX	Alinhamento do eixo X do WCS através da rotação de Frame paralelamente à orientação de ferramenta	*PGs/*	m	•	•	•	•
TOFRAMEY	Alinhamento do eixo Y do WCS através da rotação de Frame paralelamente à orientação de ferramenta	*PGs/*	m	•	•	•	•
TOFRAMEZ	como o TOFRAME	*PGs/*	m	•	•	•	•
TOLOWER	Transformação das letras de uma String em letras minúsculas	*PGAs/* Mudança para letras minúsculas / letras maiúsculas (TOLOWER, TOUPPER) (Página 80)		•	•	•	•

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
TOOLENV	Salvamento dos atuais estados importantes para a avaliação dos dados de ferramenta armazenados na memória	*FB1(W1)*		•	•	•	•
TOROT	Alinhamento do eixo Z do WCS através da rotação de Frame paralelamente à orientação de ferramenta	*PGs/*	m	•	•	•	•
TOROTOF	Rotações de Frame no sentido da ferramenta OFF	*PGs/*	m	•	•	•	•
TOROTX	Alinhamento do eixo X do WCS através da rotação de Frame paralelamente à orientação de ferramenta	*PGs/*	m	•	•	•	•
TOROTY	Alinhamento do eixo Y do WCS através da rotação de Frame paralelamente à orientação de ferramenta	*PGs/*	m	•	•	•	•
TOROTZ	como o TOROT	*PGs/*	m	•	•	•	•
TOUPPER	Transformação das letras de uma String em letras maiúsculas	*PGAs/* Mudança para letras minúsculas / letras maiúsculas (TOLOWER, TOUPPER) (Página 80)		•	•	•	•
TOWBCS	Valores de desgaste no sistema de coordenadas básico (BCS)	*PGAs/* Sistema de coordenadas da usinagem ativa (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS) (Página 411)	m	-	•	-	•
TOWKCS	Valores de desgaste no sistema de coordenadas do cabeçote da ferramenta para transformação cinemática (difere do MCS pela rotação da ferramenta)	*PGAs/* Sistema de coordenadas da usinagem ativa (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS) (Página 411)	m	-	•	-	•
TOWMCS	Valores de desgaste no sistema de coordenadas da máquina (MCS)	*PGAs/* Sistema de coordenadas da usinagem ativa (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS) (Página 411)	m	-	•	-	•

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
TOWSTD	Valor de posição inicial para correções no comprimento da ferramenta	*PGAs*/ Sistema de coordenadas da usinagem ativa (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS) (Página 411)	m	-	●	-	●
TOWTCS	Valores de desgaste no sistema de coordenadas da ferramenta (ponto de referência do porta-ferramenta T no assento do porta-ferramenta)	*PGAs*/ Sistema de coordenadas da usinagem ativa (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS) (Página 411)	m	-	●	-	●
TOWWCS	Valores de desgaste no sistema de coordenadas da peça de trabalho (WCS)	*PGAs*/ Sistema de coordenadas da usinagem ativa (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS) (Página 411)	m	-	●	-	●
TR	Componente de deslocamento em uma variável Frame	*PGAs*/ Ler e alterar componentes de Frame (TR, FI, RT, SC, MI) (Página 294)		●	●	●	●
TRAANG	Transformação de eixo inclinado	*PGAs*/ Eixo inclinado (TRAANG) (Página 380)		-	-	○	-
TRACON	Transformação concatenada	*PGAs*/ Transformações encadeadas (TRACON, TRAFOOF) (Página 396)		-	-	○	-
TRACYL	Cilindro: Transformação de superfície periférica	*PGAs*/ Transformação de superfície cilíndrica (TRACYL) (Página 371)		○	○	○	○
TRAFOOF	Desativação das transformações ativas no canal	*PGAs*/ Transformações encadeadas (TRACON, TRAFOOF) (Página 396)		●	●	●	●
TRAILOF	Movimento acoplado assíncrono de eixo OFF	*PGAs*/ Movimento acoplado (TRAILON, TRAILOF) (Página 509)		●	●	●	●
TRAILON	Movimento acoplado assíncrono de eixo ON	*PGAs*/ Movimento acoplado (TRAILON, TRAILOF) (Página 509)		●	●	●	●
TRANS	Deslocamento programável	*PGs*/	b	●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
TRANSMIT	Transformação polar (usinagem de face)	*PGAs*/ Fresamento em peças torneadas (TRANSMIT) (Página 367)		○	○	○	○
TRAORI	Transformação de 4 e 5 eixos, transformação genérica	*PGAs*/ Transformação de três, quatro e cinco eixos (TRAORI) (Página 326)		-	●	-	●
TRUE	Constante lógica: verdadeiro	*PGAs*/ Definição de variáveis de usuário (DEF) (Página 24)		●	●	●	●
TRUNC	Corte das casas decimais	*PGAs*/ Correção da precisão em erros de comparação (TRUNC) (Página 69)		●	●	●	●
TU	Ângulo do eixo	*PGAs*/ Deslocamento PTP cartesiano (Página 385)	b	●	●	●	●
TURN	Número de voltas para linha helicoidal	*PGs*/	b	●	●	●	●
ULI	Valor limite superior de variáveis	*PGAs*/ Atributo: Valores de limite (LLI, ULI) (Página 37)		●	●	●	●
UNLOCK	Liberação de ação síncrona com ID (continuação do ciclo de tecnologia)	*PGAs*/ Bloquear, liberar, resetar (LOCK, UNLOCK, RESET) (Página 646)		●	●	●	●
UNTIL	Condição para finalização de um loop REPEAT	*PGAs*/ Loop de programa com condição no início do loop (WHILE, ENDWHILE) (Página 103)		●	●	●	●
UPATH	A referência de percurso para eixos FGROUP é o parâmetro de curva	*PGAs*/ Referência ajustável do percurso (SPATH, UPATH) (Página 260)	m	●	●	●	●
VAR	Palavra-chave: Tipo de transferência de parâmetros	*PGAs*/ Chamada de subrotina com transferência de parâmetros (EXTERN) (Página 184)		●	●	●	●
VELOLIM	Redução da velocidade axial máxima	*PGAs*/ Correção percentual da velocidade (VELOLIM) (Página 499)	m	●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
VELOLIMA	Redução ou aceleração da velocidade axial máxima do eixo escravo	*PGs/*	m	●	●	●	●
WAITC	Espera até o critério de mudança de blocos de acoplamento ser preenchido para os eixos / fusos	*PGAs/* Coordenação de programa (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM) (Página 106)		-	-	○	-
WAITE	Espera pelo fim do programa em outro canal.	*PGAs/* Coordenação de programa (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM) (Página 106)		-	-	-	-
WAITENC	Espera pelas posições de eixo sincronizadas e restauradas	*PGAs/* Espera pela posição de eixo válida (WAITENC) (Página 693)		-	-	-	-
WAITM	Espera pelo marcador no canal especificado; finaliza o bloco especificado com parada exata.	*PGAs/* Coordenação de programa (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM) (Página 106)		-	-	-	-
WAITMC	Espera pelo marcador no canal especif.; parada exata somente se os outros canais ainda não alcançaram o marcador.	*PGAs/* Coordenação de programa (INIT, START, WAITM, WAITMC, WAITE, SETM, CLEARM) (Página 106)		-	-	-	-
WAITP	Espera pelo fim do deslocamento do eixo de posicionamento	*PGs/*		●	●	●	●
WAITS	Espera para alcançar a posição do fuso	*PGs/*		●	●	●	●
WALCS0	Limite da área de trabalho WCS desativado	*PGs/*	m	●	●	●	●
WALCS1	Grupo de limite de área de trabalho WCS 1 ativo	*PGs/*	m	●	●	●	●
WALCS2	Grupo de limite de área de trabalho WCS 2 ativo	*PGs/*	m	●	●	●	●
WALCS3	Grupo de limite de área de trabalho WCS 3 ativo	*PGs/*	m	●	●	●	●
WALCS4	Grupo de limite de área de trabalho WCS 4 ativo	*PGs/*	m	●	●	●	●
WALCS5	Grupo de limite de área de trabalho WCS 5 ativo	*PGs/*	m	●	●	●	●
WALCS6	Grupo de limite de área de trabalho WCS 6 ativo	*PGs/*	m	●	●	●	●

Instrução	Significado	Para descrição veja [1]	W [2]	828D [3]			
				PPU260 / 261		PPU280 / 281	
				D	F	D	F
WALCS7	Grupo de limite de área de trabalho WCS 7 ativo	*PGs/*	m	●	●	●	●
WALCS8	Grupo de limite de área de trabalho WCS 8 ativo	*PGs/*	m	●	●	●	●
WALCS9	Grupo de limite de área de trabalho WCS 9 ativo	*PGs/*	m	●	●	●	●
WALCS10	Grupo de limite de área de trabalho WCS 10 ativo	*PGs/*	m	●	●	●	●
WALIMOF	Limite da área de trabalho no BCS OFF	*PGs/*	m	●	●	●	●
WALIMON [4]	Limite da área de trabalho no BCS ON	*PGs/*	m	●	●	●	●
WHEN	A ação é executada ciclicamente enquanto a condição for preenchida.	*PGAs/* Controle cíclico da condição (WHEN, WHENEVER, FROM, EVERY) (Página 567)		●	●	●	●
WHENEVER	A ação é executada apenas uma vez quando a condição for preenchida.	*PGAs/* Controle cíclico da condição (WHEN, WHENEVER, FROM, EVERY) (Página 567)		●	●	●	●
WHILE	Início do loop de programa WHILE	*PGAs/* Loop de programa com condição no início do loop (WHILE, ENDWHILE) (Página 103)		●	●	●	●
WRITE	Gravação de texto no sistema de arquivos. Anexa um bloco no fim do arquivo especificado.	*PGAs/* Gravar arquivo (WRITE) (Página 132)		●	●	●	●
WRTPR	Retardamento da tarefa de usinagem sem interromper o modo de controle da trajetória	*PGAs/*		●	●	●	●
X	Nome de eixo	*PGs/*	m/b	●	●	●	●
XOR	OU lógico exclusivo	*PGAs/* Operadores de comparação e operadores lógicos (Página 67)		●	●	●	●
Y	Nome de eixo	*PGs/*	m/b	●	●	●	●
Z	Nome de eixo	*PGs/*	m/b	●	●	●	●

Apêndice

A

A.1 Lista de abreviações

A	Saída
AS	Sistema de automação
ASCII	American Standard Code for Information Interchange: Norma americana de códigos para troca de informações
ASIC	Application Specific Integrated Circuit: Circuito de aplicação do usuário
ASUP	Subrotina assíncrona
AV	Preparação do trabalho
AWL	Lista de instruções
BA	Modo de operação
BAG	Grupo de modos de operação
BB	Pronto para operar
BuB, B&B	Operar e observar
BCD	Binary Coded Decimals: Números decimais codificados em código binário
BHG	Terminal portátil
BIN	Arquivos binários (**Bin**ary Files)
BIOS	Basic Input Output System
BCS	Sistema de coordenadas básico
BOF	Interface de operação
BOT	Boot Files: Arquivos de boot para SIMODRIVE 611 digital
BT	Painel de comando
BTSS	Interface de painel de comando
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CNC	Computerized Numerical Control: Comando numérico computadorizado
COM	Communication
CP	Communication Processor (Processador de comunicação)
CPU	Central Processing Unit: Unidade de processamento central
CR	Carriage Return
CRT	Cathode Ray Tube: Tubos de raios catódicos
CSB	Central Service Board: Unidade de PLC
CTS	Clear To Send: Mensagem de pronto para enviar em interfaces de dados seriais
CUTCOM	Cutter radius compensation: Correção do raio da ferramenta
DAU	Conversor digital-analógico
DB	Módulo de dados no PLC
DBB	Byte de módulo de dados no PLC
DBW	Palavra de módulo de dados no PLC

DBX	Bit de módulo de dados no PLC
DC	Direct Control: Movimento do eixo rotativo pelo curso mais curto até a posição absoluta realizado durante uma rotação
DCD	Carrier Detect
DDE	Dynamic Data Exchange
DEE	Dispositivo terminal de dados
DIN	Deutsche Industrie Norm (Norma industrial alemã)
DIO	Data Input/Output: Exibição da transferência de dados
DIR	Directory: Diretório
DLL	Dynamic Link Library
DOE	Dispositivo de transferência de dados
DOS	Disk Operating System
DPM	Dual Port Memory
DPR	Dual-Port-RAM
DRAM	Dynamic Random Access Memory
DRF	Differential Resolver Function: Função de resolução diferencial (manivela eletrônica)
DRY	Dry Run: Avanço de teste
DSB	Decoding Single Block: Bloco a bloco de decodificação
DW	Palavra de dados
E	Entrada
E/A	Entrada/saída
E/R	Unidade de alimentação e realimentação (de tensão) do SIMODRIVE 611 digital
Código EIA	Código especial de fita perfurada, o número de furos por caractere é sempre ímpar
ENC	Encoder: Gerador de valor real
EPROM	Erasable Programmable Read Only Memory (memória de leitura deletável e eletricamente programável)
ERROR	Error from printer
FB	Módulo de função
FBS	Tela plana
FC	Function Call: Módulo de função no PLC
FDB	Banco de dados do produto
FDD	Floppy Disk Drive
FEPROM	Flash-EPROM: Memória de leitura e gravação
FIFO	First In First Out: Memória, que opera sem indicação de endereço e cujos dados podem ser lidos na mesma sequência em que vão sendo armazenados.
FIPO	Interpolador fino
FM	Módulo de função
FPU	Floating Point Unit: Unidade de ponto flutuante
FRA	Módulo do Frame
FRAME	Bloco de dados (quadro)
FRK	Correção do raio da fresa
FST	Feed Stop: Parada de avanço
FUP	Plano de funcionamento (método de programação para PLC)

GP	Programa básico
GUD	Global User Data: Dados de usuário globais
HD	Hard Disk: Disco rígido
HEX	Abreviação para número hexadecimal
HiFu	Função auxiliar
HMI	Human Machine Interface: Funcionalidade de operação do SINUMERIK para operação, programação e simulação.
HMS	Sistema de medição de alta resolução
HSA	Acionamento do fuso principal
HW	Hardware
IBN	Colocação em funcionamento
IF	Habilitação de pulsos do módulo de acionamento
IK (GD)	Comunicação implícita (dados globais)
IKA	Interpolative Compensation: Compensação interpolatória
IM	Módulo de interface: Módulo de interface
IMR	Interface-Modul Receive: Módulo de interface para modo de recepção
IMS	Interface-Modul Send: Módulo de interface para modo de envio
INC	Increment: Incremento
INI	Initializing Data: Dados de inicialização
IPO	Interpolador
ISA	International Standard Architecture
ISO	International Standard Organization
Código ISO	Código especial de fita perfurada, o número de furos por caractere é sempre par
JOG	Jogging: Modo de ajuste
K1 .. K4	Canal 1 até canal 4
K-Bus	Bus de comunicação
KD	Rotação de coordenadas
KOP	Plano de contatos (método de programação para PLC)
K_v	Fator de amplificação do circuito
$K_Ü$	Relação de transmissão
LCD	Liquid-Crystal Display: Display de cristal líquido
LED	Light-Emitting Diode: Diodo emissor de luz
LF	Line Feed
LMS	Sistema de medição de posição
LR	Controlador de posição
LUD	Local User Data
MB	Megabyte
MD	Dados de máquina
MDA	Manual Data Automatic: Entrada manual
MK	Circuito de medição
MCS	Sistema de coordenadas da máquina
MLFB	Denominação de produto legível por máquina
MPF	Main Program File: Programa de peça do NC (programa principal)

MPI	Multi Port Interface: Interface multiponto
MS-	Microsoft (fabricante de software)
MSTT	Painel de comando da máquina
NC	Numerical Control: Comando numérico
NCK	Numerical Control Kernel: Núcleo numérico com preparação de blocos, área de deslocamento, etc.
NCU	Numerical Control Unit: Unidade de hardware do NCK
NRK	Denominação do sistema operacional do NCK
NST	Sinal de interface
NURBS	Non-Uniform Rational B-Spline
NV	Deslocamento de ponto zero
OB	Módulo de organização no PLC
OEM	Original Equipment Manufacturer
OP	Operation Panel: Painel de operação
OPI	Operation Panel Interface: Interface do painel de comando
OPT	Options: Opcionais
OSI	Open Systems Interconnection: Norma para comunicação do processador
P-Bus	Bus periférico
PC	Personal Computer
PCIN	Nome do SW para troca de dados com o comando
PCMCIA	Personal Computer Memory Card International Association: Padrão para cartões de memória
PCU	PC Unit: PC-Box (unidade de processamento)
PG	Dispositivo de programação
PLC	Programmable Logic Control: Controle de adaptação (programável)
POS	De posicionamento
RAM	Random Access Memory: Memória de programa que pode ser lida e gravada
REF	Função de aproximação do ponto de referência
REPOS	Função de reposicionamento
RISC	Reduced Instruction Set Computer: Tipo de processador de bloco de comandos reduzido e rápido processamento dos comandos
ROV	Rapid Override: Correção de avanço rápido
RPA	R-Parameter Active: Área de memória no NCK para R-NCK e números de parâmetro R
RPY	Roll Pitch Yaw: Tipo de rotação de um sistema de coordenadas
RTS	Request To Send: Ativa uma parte de envio, sinal de controle de interfaces seriais de dados
SBL	Single Block: Bloco a bloco
SD	Dado de ajuste
SDB	Módulo de dados de sistema
SEA	Setting Data Active: Identificação (tipo de arquivo) para dados de ajuste
SFB	Módulo de função do sistema
SFC	System Function Call
SK	Softkey

SKP	Skip: Salto (omissão) de bloco
SM	Motor de passo
SPF	Sub Program File: Subrotina
SPS	Comando lógico programável
SRAM	Memória estática (armazenada)
SRK	Correção do raio de corte
SSFK	Compensação de erro de passo do fuso
SSI	Serial Synchron Interface: Interface serial síncrona
SW	Software
SYF	System Files: Arquivos de sistema
TEA	Testing Data Active: Identificação para dados de máquina
TO	Tool Offset: Correção de ferramenta
TOA	Tool Offset Active: Identificação (tipo de arquivo) para correções de ferramenta
TRANSMIT	Transform Milling into Turning: Conversão de coordenadas em tornos para operações de fresamento
UFR	User Frame: Deslocamento de ponto zero
UP	Subrotina
VSA	Acionamento de avanço
V.24	Interface serial (definição dos cabos de troca entre DEE e DÜE)
WCS	Sistema de coordenadas da peça de trabalho
WKZ	Ferramenta
WLK	Correção do comprimento da ferramenta
WOP	Programação orientada para oficinas
WPD	Work Piece Directory: Diretório de peças de trabalho
WRK	Correção do raio da ferramenta
WZK	Correção de ferramenta
WZW	Troca de ferramentas
ZOA	Zero Offset Active: Identificação (tipo de arquivo) para dados de deslocamento de ponto zero
µC	Micro-controlador

A.2 Feedback sobre a documentação

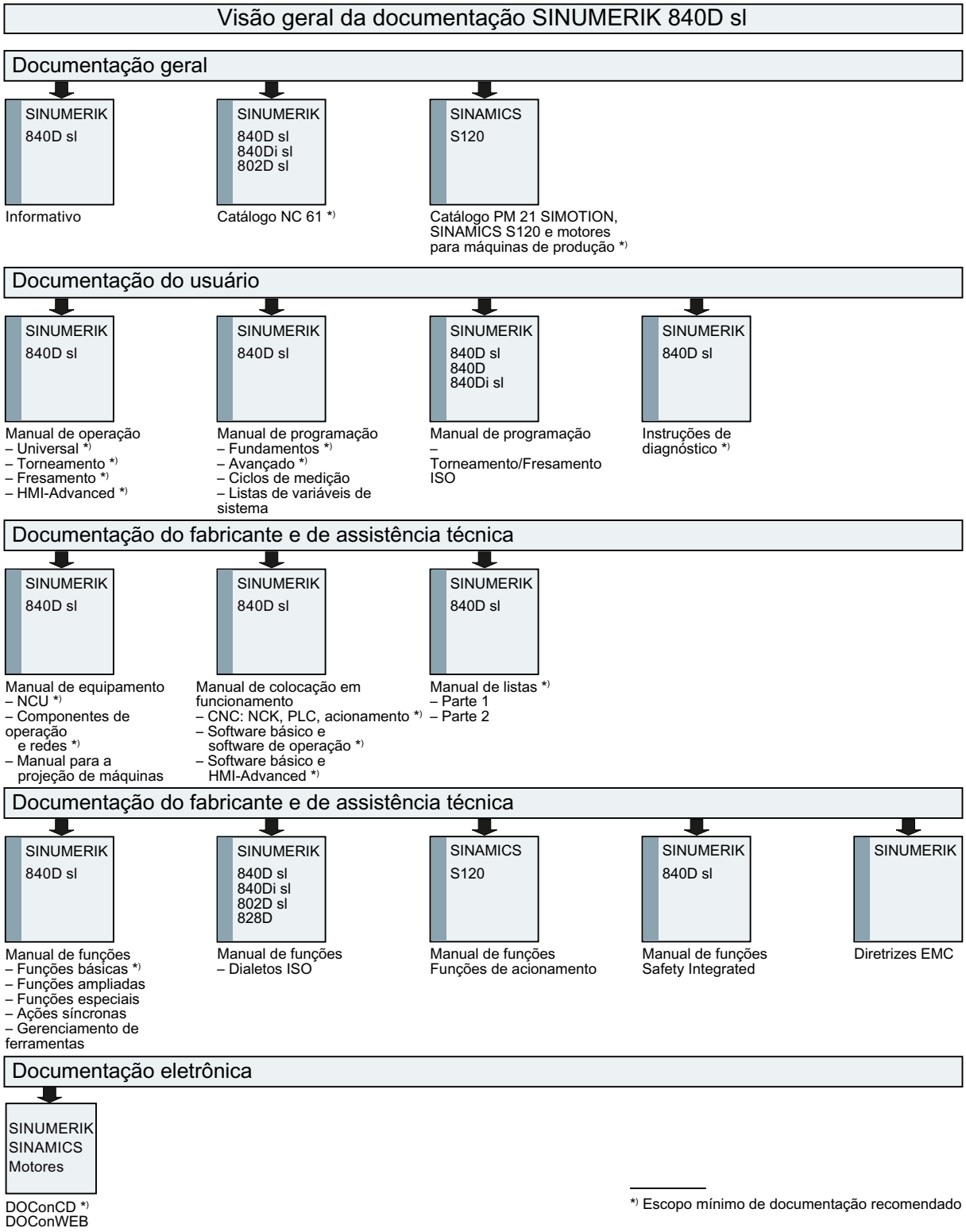
O presente documento vem sendo continuamente aprimorado em qualidade e em satisfação do usuário. Por favor, colabore conosco mencionando suas observações e sugestões de melhoria enviando um E-Mail ou FAX para:

E-Mail:	mailto:docu.motioncontrol@siemens.com
Fax:	+49 9131 - 98 2176 Utilize o modelo de FAX disponível no verso da folha.

A SIEMENS AG I DT MC MS1 Postfach 3180 D-91050 Erlangen Fax.: +49 9131 - 98 2176 (Documentação)	Remetente Nome: Endereço da empresa/departamento Rua: CEP: Cidade: Telefone: / Telefax: /

Sugestões e/ou correções

A.3 Vista Geral da documentação


Visão geral da documentação SINUMERIK 840D sl
Documentação geral
SINUMERIK 840D sl
Informativo
SINUMERIK 840D sl 840Di sl 802D sl
Catálogo NC 61 *)
SINAMICS S120
Catálogo PM 21 SIMOTION, SINAMICS S120 e motores para máquinas de produção *)
Documentação do usuário
SINUMERIK 840D sl
Manual de operação – Universal *) – Torneamento *) – Fresamento *) – HMI-Advanced *)
SINUMERIK 840D sl
Manual de programação – Fundamentos *) – Avançado *) – Ciclos de medição – Listas de variáveis de sistema
SINUMERIK 840D sl 840D 840Di sl
Manual de programação – Torneamento/Fresamento ISO
SINUMERIK 840D sl
Instruções de diagnóstico *)
Documentação do fabricante e de assistência técnica
SINUMERIK 840D sl
Manual de equipamento – NCU *) – Componentes de operação e redes *) – Manual para a projeção de máquinas
SINUMERIK 840D sl
Manual de colocação em funcionamento – CNC: NCK, PLC, acionamento *) – Software básico e software de operação *) – Software básico e HMI-Advanced *)
SINUMERIK 840D sl
Manual de listas *) – Parte 1 – Parte 2
Documentação do fabricante e de assistência técnica
SINUMERIK 840D sl
Manual de funções – Funções básicas *) – Funções ampliadas – Funções especiais – Ações síncronas – Gerenciamento de ferramentas
SINUMERIK 840D sl 840Di sl 802D sl 828D
Manual de funções – Dialetos ISO
SINAMICS S120
Manual de funções Funções de acionamento
SINUMERIK 840D sl
Manual de funções Safety Integrated
SINUMERIK
Diretrizes EMC
Documentação eletrônica
SINUMERIK SINAMICS Motores
DOConCD *) DOConWEB

*) Escopo mínimo de documentação recomendado

Glossário

Aceleração com limitação de torque

Para otimizar a resposta de aceleração na máquina, e simultaneamente proteger a mecânica, pode-se alternar no programa de usinagem entre a aceleração rápida e a aceleração constante (sem jerk).

Acionamento

O acionamento é o componente do CNC que controla o torque e a rotação do motor baseado em comandos do NC.

Ações sincronizadas

1. Emissão de função auxiliar

   Durante a usinagem de uma peça pode-se solicitar funções tecnológicas externas (→ Funções auxiliares) do programa de CNC ao CLP. Por exemplo, estas funções auxiliares são utilizadas para controlar equipamentos auxiliares da máquina-ferramenta, como mandril, garras de fixação, porta-ferramenta, etc.

2. Apresentação de funções rápidas de ajuda

   Com relação ao tempo crítico de alteração de funções, o tempo de reconhecimento para as → funções auxiliares pode ser minimizado e paradas desnecessárias no processo de usinagem são evitados.

Alarmes

Todas → Mensagens e alarmes são indicados no painel de operação com data e hora, e o símbolo correspondente para indicar o critério de eliminação. Alarmes e mensagens são mostrados separadamente.

1. Alarmes e mensagens em programas de usinagem

   Alarmes e mensagens podem ser geradas diretamente de programas de usinagem.

2. Alarmes e mensagens do PLC

   Alarmes e mensagens de máquina podem ser geradas pelo programa de PLC. Para isso nenhum pacote adicional de blocos de função é necessário.

Aproximação de ponto fixo

Máquina-ferramenta pode definir pontos fixos para troca de ferramenta, carregamento, troca de paletes, etc. As coordenadas para estes pontos são armazenadas no comando. O comando movimenta os eixos envolvidos, se possível, em → avanço rápido.

Área de offset de ferramenta ativa (TOA)

Na área de offset de ferramenta ativa contém todos os dados de ferramenta e magazine. Por padrão, esta área coincide com a área do → canal considerando o objetivo dos dados. No entanto, os dados de máquina podem ser utilizados para permitir que vários canais compartilhem de uma unidade → TOA, então dados comuns do gerenciador de ferramentas ficam disponíveis para estes canais.

Área de trabalho

Área tri-dimensional, na qual a ponta da ferramenta pode se mover, com base na construção da máquina-ferramenta. Vide → Área de proteção.

Arquivar

Transmissão de arquivos ou diretórios para um dispositivo **externo** de armazenamento.

Aterramento

No terra é conectado todas as partes inativas de um equipamento, o qual mesmo em caso de mal funcionamento não se tornará ativa gerando risco de contato com alguma tensão.

Automático

Modo de operação do comando (Operação em sequência de blocos de acordo com a DIN): Modo de operação do sistema NC, em que um → Programa de usinagem é selecionado e processado de forma contínua.

Avanço de tempo inverso

No SINUMERIK 840D pode ser programado o tempo necessário para o deslocamento pelo trajeto indicado em um bloco para a movimentação do eixo ao invés da velocidade de avanço (G93).

Avanço de trajetória

Avanço de trajetória influência → eixos de trajetória. Ele representa a soma geométrica dos avanços dos → eixos geométricos envolvidos.

Avanço rápido

Avanço mais rápido de um eixo. É utilizado quando, por exemplo, a ferramenta está se aproximando de um → contorno da peça de uma posição de descanso ou está sendo recuada. O avanço rápido é definido em uma base de máquina específica através de um dado de máquina.

Bateria reserva

A bateria reserva garante que → o programa de usuário na → CPU será protegido de falhas na alimentação e mantém fixas as áreas de dados e indicadores, temporizadores e contadores.

Bloco

Todas as configurações para as necessidades programação e execução dos programas são realizadas nos blocos.

Bloco de dados

1. A unidade de dados do → PLC, que pode acessar → programas HIGHSTEP
2. Unidade de dados do → NC: Bloco de dados que contém definições de dados para usuários globais. Os dados podem ser inicializados diretamente em sua configuração.

Bloco de programa

Bloco de programa contém o programa principal e sub-rotinas do → programa de peça.

Bloco principal

Um bloco antecedido por ":" bloco introdutório, contém todos os parâmetros necessários para iniciar a execução de um → programa de usinagem.

Bloco secundário

Bloco introduzido por "N" com informação sobre a etapa do processo, por exemplo, um dado de posição.

Blocos intermediários

Operação de movimentação com a seleção → de compensação de ferramenta (G41/G42) pode ser interrompida por uma limitação na quantidade de blocos intermediários (Bloco sem movimentação de eixo no plano de compensação), de forma que a compensação de ferramenta ainda possa ser corretamente realizada. A quantidade de blocos intermediários permitidos, que o comando lê antecipadamente, é ajustável através dos parâmetros de sistema.

Boot

Carrega os programas de sistema ao ligar.

Cabo de conexão

Cabos de conexão são pré-fabricados ou podem ser montados pelo usuário, os cabos tem dois fios com um conector em cada ponta. Este cabo de conexão conecta a → CPU através da → interface MPI (multi-point interface) com uma → PG ou com outras CPUs.

Canal

Um canal é caracterizado pelo fato de que um → programa de usinagem pode ser processado independentemente de outros canais. Um canal controla exclusivamente os eixos e fusos associados à ele. Programas de usinagem trabalham em canais diferentes podem ser coordenados através de → sincronização.

Canal de execução

A estrutura do canal pode ser utilizada para redução de tempo não produtivo através de sequências de movimentos em paralelo, por exemplo, movimento em uma porta de carregamento simultâneo à usinagem. Um canal de CNC deve ser considerado como um comando de CNC separado, com decodificação, preparação de bloco e interpolação.

Chaves

As chaves no → painel de comando da máquina possui quatro posições, as quais possuem funções definidas no sistema de operação do comando. Ao interruptor das chaves são associadas três chaves de cores diferentes, que podem ser movimentadas para as posições específicas.

Chaves de programação

Caracteres ou strings, que possuem um significado fixo na linguagem de programação do → programa de usinagem.

Ciclos

Subrotinas protegidas para suporte na realização de usinagens repetitivas em uma → peça.

Ciclos padrão

Para tarefas de usinagem utilizadas frequentemente são disponíveis os ciclos padrão:

- para furação/fresamento
- para torneamento

Na área de operação "Programa" no menu "Ciclos de auxílio" são listados os ciclos disponíveis. Após a seleção do ciclo desejado são apresentados parâmetros necessários para preenchimento.

CLP

Controle **L**ógico **P**rogramável: → Controlador lógico programável. Componente do → NC: Controlador programável para processar o controle lógico da máquina-ferramenta.

CNC

Vide → NC

COM

Componente do comando de NC para realização e coordenação de comunicação.

Compensação de erro no passo do fuso

Compensação de irregularidades mecânicas em um avanço de todas as peças relacionadas ao fuso esférico com base nas variações dos valores medidos.

Compensação de folga

Compensação de folga mecânica da máquina, por exemplo folga no fuso esférico reverso. Para cada eixo a compensação de folga deve ser especificada separadamente.

Compensação do erro de quadrante

Erro de contorno na transição de quadrantes, que aumenta pela alteração no atrito das guias de condução, pode ser virtualmente eliminado com a compensação de erro de quadrante. A parametrização da compensação de erro de quadrante ocorre através de um teste circular.

Compensação do raio da ferramenta

Visando a programação do → contorno de peça desejado, o comando deve percorrer o contorno programado em uma trajetória eqüidistante considerando o raio da ferramenta utilizada, (G41/G42).

Compensação interpolatória

Compensação interpolatória é uma ferramenta que habilita o erro de controle de fuso (**S**pindel**s**teigungs**f**ehler) e o erro de compensação do sistema de medição (**M**ess**s**ystem**f**ehler **k**ompensiert) do fabricante (SSFK, MSFK).

Contorno

Forma da → Peça

Contorno acabado

Contorno da peça usinada. Vide → Peça bruta.

Contorno da peça de trabalho

Contorno desejado para → peças a serem criadas/usinadas.

Controle de avanço dinâmico

Irregularidades no contorno devido à erros de contorno podem ser praticamente eliminadas utilizando o controle de avanço dependente da aceleração. Isto resulta em uma excelente precisão de usinagem mesmo em altas → velocidades. O controle pode selecionar ou retirar a seleção de um eixo específico no → programa de peça.

Controle de velocidade

Visando a obtenção de uma velocidade aceitável em caso de movimentos irrelevantes por bloco, uma análise antecipada em vários blocos pode ser realizada (→ Look Ahead).

Controle Lógico Programável

Controles programáveis (CLP) são controles eletrônicos, no qual suas funções são armazenadas em forma de programa na unidade de controle. A estrutura e a fiação do equipamento não dependem da função do controlador. O controlador programável tem a mesma estrutura que um computador; ele consiste em uma CPU (unidade central) com memória, grupo de entradas e saídas e um bus-system interno. Os periféricos e a linguagem de programação estão alinhados aos interesses do controle.

Coordenadas polares

Sistema de coordenadas, que especifica o local do ponto no plano através de sua distância do ponto zero e o ângulo formado pelo vetor de direção com o eixo fixo.

Corretor do raio de corte

Através da programação de um contorno a ponta da ferramenta é desconsiderada. Na prática isto não é realizado, o raio da ferramenta selecionada deve ser indicado no controle e é considerado desta forma. Desta forma o centro do raio é deslocado de forma eqüidistante do contorno.

Corretores de ferramenta

Consideração das dimensões da ferramenta para o cálculo do trajeto.

CPU

Central Processing Unit, vide → Controle programável

C-Spline

O C-Spline é o Spline mais conhecido e utilizado. O trajeto pelos pontos base são tangentes e com curvatura constante. É utilizado um polinômio de terceiro grau.

Curvatura

A curvatura k de um contorno é o inverso do raio r da aproximação em círculo em um ponto de contorno (k = 1/r).

Dados de ajuste

Dados, que comunica propriedades da máquina-ferramenta ao NC, como foi definida pelo Systemsoftware.

Definição de variáveis

Na definição de uma variável contém o tipo de dado e o nome da variável. Com o nome da variável pode-se endereçar o valor desta.

Deslocamento de ponto zero

Padronizar um novo ponto de referência para um sistema de coordenadas através da aquisição de um ponto zero existente e um → frame.

1. Ajuste

   SINUMERIK 840D: Existe uma certa quantidade de deslocamentos de zero ajustáveis para cada eixo CNC à disposição. Os deslocamentos, que são selecionados através de funções G, são ativados opcionalmente.

2. Externo

   Adicionalmente para todos deslocamentos, que determinam a localização do ponto zero da peça, um deslocamento de zero externo pode ser sobreposto através da manivela eletrônica (deslocamento DRF) ou através do PLC.

3. Programável

   Com a instrução `TRANS` pode-se programar o deslocamento de ponto zero para todos os eixos de posicionamento e trajetória.

Deslocamento externo de ponto zero

Deslocamento de ponto zero especificado pelo → PLC.

Diagnóstico

1. Área de operação do controle
2. O controle possui tanto um auto-diagnóstico quanto um teste auxiliar para o trabalho: Indicações de estado, alarme e trabalho.

Dimensão absoluta

O destino para movimento de um eixo é definido por cotas que se referem ao sistema de coordenadas atualmente ativo. Vide → Sequência de medição.

Dimensão incremental

Também medidas incrementais: O destino de um eixo transversal é definido através de uma distância e direção orientadas a partir de um ponto já alcançado. Vide → medição absoluta.

DRF

Differential Resolver Function: Funções de NC, que gera um deslocamento de ponto zero incremental em modo automático com utilização de uma manivela eletrônica.

Editor

O editor permite a criação, alteração, complementação, junção e inserção de programas/textos/blocos.

Editor de texto

Vide → Editor

Eixo base

Eixos, para os quais o valor de referência ou o valor atual de posição gera uma base de cálculo para um valor de compensação.

Eixo C

Eixo, ao redor do qual a ferramenta do fuso descreve uma rotação e movimentos de posicionamento controlados.

Eixo de compensação

Eixo, o qual o valor atual ou desejado é alterado de acordo com o valor de compensação.

Eixo de posicionamento

Eixo, que realiza um movimento auxiliar na máquina-ferramenta. (por exemplo, Magazine de ferramentas, transporte de paletes). Eixos de posicionamento são eixos, que não interpolam com → os eixos de trajetória.

Eixo de sincronismo

O eixo de sincronismo é o → eixo gantry, cujas posições desejadas derivam continuamente dos movimentos do → eixo mestre, e move-se de forma sincronizada à este. À vista do operador e do programador, o eixo de sincronismo "não é presente".

Eixo de trajetória

Eixos de trajetória são todos os eixos processados pelo → canal que são controlados pelo → interpolador de forma que iniciam, aceleram, param e atingem o ponto final simultaneamente.

Eixo geométrico

Eixos geométricos servem para descrever planos em duas ou três dimensões em um sistema de coordenadas da peça.

Eixo linear

O eixo linear descreve uma linha reta diferente do eixo rotativo.

Eixo mestre

O eixo mestre é o → eixo gantry, que existe do ponto de vista do operador e do programador e por isso é manipulado como um eixo de NC padrão.

Eixo rotativo

Eixos rotativos produzem um giro da peça ou da ferramenta de acordo com um ângulo especificado.

Eixos

De acordo com suas funções, os eixos de CNC são classificados como:

- Eixos: eixos de interpolação de trajetória
- Eixos auxiliares: Eixos de posicionamento sem interpolação com avanço programado individualmente. Eixos auxiliares não participam de usinagem, Ex. Trocador de ferramentas, Magazine de ferramentas.

Eixos de curvatura

Eixos curvatura produzem um giro da peça ou da ferramenta de acordo com um passo definido. Ao atingir o passo definido, o eixo está "posicionado".

Eixos de máquina

Eixos físicos existentes em uma máquina-ferramenta.

Eixos sincronizados

Eixos síncronos necessitam para seu trajeto o mesmo tempo que um eixo geométrico necessita.

Endereço

Um endereço é o identificador para um certo operando ou faixa de operandos, Ex entrada, saída etc.

Endereço de eixo

Vide → Identificador de eixo

Escala

Componentes de um → frame, que causa variações de escala.

Especificação de dimensão métrica e em polegadas

No programa de usinagem podem ser programados valores de posições ou passos em polegadas. Independentemente da indicação de medição do programa (G70/G71) o controle é ajustado em um sistema base.

Espelhamento

Através do espelhamento os valores das coordenadas de um contorno são alteradas de forma oposta com relação à um eixo. Pode ser espelhado em vários eixos ao mesmo tempo.

Faixa de deslocamento

A faixa máxima de deslocamento permitida para um eixo linear é de ± 9 décadas. O valor absoluto depende da entrada selecionada, da resolução de controle de posição e do sistema de medição (polegadas ou metro).

Ferramenta

Peça ativa na máquina-ferramenta, que realiza a usinagem (por exemplo, ferramenta de corte, fresa, broca, feixe de laser ...).

Fim de curso de Software

Fim de curso de software limita a faixa de deslocamento de um eixo e previne colisões da mecânica nos limites de hardware. Para cada eixo existe um par, que pode ser ativado separadamente através → do CLP.

Frame

Um frame é representado por uma fórmula aritmética, que transfere um sistema de coordenadas cartesianas para outro sistema de coordenadas cartesianas. Um frame contém as seguintes funções → deslocamento de ponto zero, → rotação, → alteração de escala, → espelhamento.

Frames programáveis

Os → frames programáveis permitem a definição dinâmica de novos pontos de início do sistema de coordenadas durante a execução do programa de usinagem. É utilizado uma definição absoluta com frames novos ou uma definição adicional referente à um ponto de início existente.

Funções auxiliares

Funções auxiliares permite ao programa de usinagem transferir → parâmetros ao → CLP que podem disparar reações definidas pelo fabricante de máquina.

Funções de segurança

O controle contém um monitoramento constante, que detecta falhas no → CNC, no → CLP e na máquina de maneira que é amplamente prevenida alguma danificação das peças, ferramentas ou da máquina. Em caso de falha, a usinagem é interrompida e os acionamentos são parados, a causa do mau funcionamento é armazenada e o alarme é apresentado. Simultanemante, o CLP é informado que um alarme de CNC é apresentado.

Geometria

Descrição de uma → peça em → um sistema de coordenada de peça.

Gerenciamento de programas de usinagem

O gerenciamento de programas de usinagem pode ser organizado por → peça. O tamanho da memória determina o número de programas e dados que poderão ser gerenciados. Cada arquivo (programa ou dado) pode ter um nome com no máximo 24 caracteres alfa numéricos.

Grupo de modos de operação

Eixos e fusos que são tecnologicamente acoplados podem ser combinados em um mesmo grupo de operação (BAG). Eixos/Fusos de um BAG podem ser controlados por um ou vários → canais. O mesmo → modo de operação é sempre atribuído aos canais do BAG.

HIGHSTEP

Sumário para as possibilidades de programação para o → PLC do sistema AS300/AS400.

Identificador

De acordo com a DIN 66025, palavras são complementadas utilizando indicadores (nomes) para variáveis (variáveis de cálculo, variáveis de sistema, variáveis de usuário), para sub-rotinas, palavras-chaves e palavras com várias letras de endereçamentos suplementares. Este complemento tem o mesmo significado das palavras respeitando a construção do bloco. Os identificadores devem ser únicos. O mesmo identificador não pode ser utilizado por diferentes objetos.

Identificador de eixo

Os eixos são identificados como X, Y e Z de acordo com a DIN 66217, para sistema de coordenadas obedecendo as regras da mão direita.

Eixo rotativo em torno de X, Y e Z são identificados como A, B, C. Outros eixos paralelos aos indicados, podem ser identificados por outras letras.

Incremento

Indicação de distância do movimento de acordo com o valor do incremento. Valor do incremento pode ser definido pelos → dados de ajuste e/ou selecionado através das teclas 10, 100, 1000, 10000.

Interface de operação

A interface homem-máquina (IHM) é um indicador do comando CNC com auxilio de telas. É composta por softkeys horizontais e verticais.

Interface serial V.24

Para a entrada/saída de dados na PCU 20 existe uma interface serial V.24 (RS232), já para a PCU50/70 existem 2 interfaces seriais disponíveis. Através desta interface podem ser carregados ou salvos tanto programas de usinagem quanto dados de máquina de fabricante e usuário.

Interpolação circular

A → ferramenta deve movimentar-se em círculo entre pontos definidos do contorno com um avanço estipulado e então a peça é usinada.

Interpolação de polinômios

Com a interpolação de polinômios os trajetos e curvas mais variados podem ser gerados, como **funções lineares, parábolas, funcões exponenciais** (SINUMERIK 840D).

Interpolação de Spline

Com a interpolação Spline o controle pode gerar uma curva característica bem definida, com apenas alguns pontos base.

Interpolação helicoidal

A interpolação helicoidal é apropriada particularmente à usinagem de rosca interna ou externa com fresa para chanfro e para fresamento de ranhuras de lubrificação.

O movimento helicoidal consiste em dois movimentos em conjunto:

- Movimento circular em um plano
- Movimento linear perpendicular à este plano

Interpolação linear

A ferramenta irá se movimentar por uma linha reta até o destino enquanto usina a peça.

Interpolador

Unidade lógica do → NCK, a qual determina valores intermediários para o movimento, a ser realizado em eixos individuais com base na informação de posições finais especificadas no programa de usinagem.

JOG

Modo de operação do controle (Ajuste modo de operação): A máquina pode ser ajustada no modo de operação JOG. Eixos individuais e fusos podem ser movimentados em JOG através das teclas de direção. Outras funções para o modo de operação JOG são: → referenciamento, → Repos e → Preset (ajuste de posição atual).

KV

Fator de ganho do servo, variável de controle em uma malha fechada.

Limite de área de trabalho

Com o auxílio da limitação da área de trabalho, o deslocamento dos eixos pode ser limitado além das chaves fim de curso. Cada eixo possui um par de valores para definição da área de trabalho protegida.

Limite de parada exata.

Quando todos os eixos atingem o limite de parada exata, o controlador se comporta como se tivesse atingido seu ponto exato de destino. Ocorre um avanço de bloco no → programa de peça.

Limite de Velocidade

Velocidade máxima/mínima do fuso: Através dos dados de máquina, o → PLC ou → os dados de configuração podem limitar a rotação máxima do fuso.

Limite programável da área de trabalho

Limitação da área de movimentação da ferramenta através da programação de limites da área definida.

Linguagem alto-nível do CNC

A linguagem de alto nível oferece: → Variáveis de usuário, → Variáveis de sistema, → Tecnologia de macros.

Look Ahead

Com a função **Look Ahead** consegue-se otimizar a velocidade de usinagem, através da visualização antecipada de uma certa quantidade de blocos.

MDA

Modo de operação do comando: Manual Data Automatic. No modo de operação MDA, blocos de programa individuais ou seqüência de blocos, sem ter referência à um programa ou sub-rotina, podem ser definidos e instantaneamente executados pela tecla NC-Start.

Memória de carregamento

A memória de carregamento é igual à → RAM para a CPU 314 do → CLP.

Memória de compensação

Área de dados do comando, onde são armazenados os dados de corretores de ferramenta.

Memória de programação de CLP

SINUMERIK 840D: O programa de usuário, dados de usuários e o programa base de CLP são armazenados juntos na memória de usuário do CLP.

Memória de sistema

A memória de sistema é uma memória da CPU, onde os seguintes dados são arquivados:

- Dados, que são requeridos pelo sistema
- Os operandos de tempo, contador, indicador

Memória de trabalho

A memória de trabalho é uma memória RAM dentro da → CPU, que o processador acessa durante a execução do programa de usuário.

Memória de usuário

Todos os programas e dados como programas de usinagem, sub-rotinas, comentários, correção de ferramenta, deslocamento de ponto zero/frames, assim como, dados de usuário de programa e canal podem ser armazenados na memória comum de usuário do CNC.

Mensagens

Todas as mensagens programadas em um programa de usinagem e → alarmes detectados pelo sistema são indicados no painel de operação com data e hora e com o símbolo correspondente para seu cancelamento. A indicação de alarmes e mensagens são apresentadas separadamente.

Modo contínuo de trajetória

A função do Modo contínuo de trajetória é evitar desacelerações substanciais → dos eixos de trajetória nas fronteira entre blocos do programa de usinagem e continuar com o avanço o mais próximo possível no bloco seguinte.

Modo de operação

Conceito de operação do comando SINUMERIK. Os seguintes modos são definidos: → Jog → MDA → Automático.

Módulo periférico

Módulos I/O realizam a conexão entre CPU e o processo.

Módulos I/O são:

- → Módulos de entrada/saída digital
- → Módulos de entrada/saída analógica
- → Módulos de simulação

Monitoração de contorno

O erro de contorno é monitorado considerando-se uma faixa de valores de tolerância pré-definidos como precisão do contorno. Um erro de contorno ilegal pode causar, por exemplo, do sobrecarregamento do acionamento. Neste caso apresentará um alarme e o eixo será parado.

NC

Numerical Control: Comando numérico (NC) contém todos os componentes do controle da máquina-ferramenta: → NCK, → PLC, HMI, → COM.

Indicação

Um termo mais apropriado para o comando SINUMERIK 840D seria: Computerized Numerical Control.

NCK

Numerical Control Kernel: Componente do comando de NC que executa o → programa de usinagem e as coordenadas base das operações de movimentação para a máquina-ferramenta.

Nome de eixo

Vide → Identificador de eixo

NRK

Numeric Robotic Kernel (Sistema operacional → NCK)

NURBS

O controle de movimentação e interpolação de trajetória que ocorre no comando é feito com base em NURBS (**N**on **U**niform **R**ational **B**-**S**plines). Como resultado, um movimento uniforme é disponível no comando para todas as interpolações do SINUMERIK 840D.

OEM

O escopo para implementação de soluções individuais (aplicações OEM) para SINUMERIK 840D é fornecido pelos fabricantes de máquina que desejem criar sua própria IHM ou incluir funções de processo específicas no comando.

Override

Controle manual ou programável, que permite ao operador alterar avanços programáveis ou rotações, de acordo com a peça ou material.

Override de avanço

A velocidade programada é sobreposta pelo ajuste da velocidade atual feita através → do painel de comando da máquina ou pelo → PLC (0-200%). O avanço de velocidade pode igualmente ser corrigido no programa de usinagem através de uma faixa de porcentagem (1-200%).

Painel de comando da máquina

Painel de operação da máquina-ferramenta com teclas de controle, potenciômetros, etc, e um indicador simples com LEDs. Ele serve para interagir diretamente com a máquina-ferramenta, através do PLC.

Palavra de dados

Unidade de dados de dois bytes dentro de um → bloco de dados.

Palavras-chave

Palavras com sintaxe definida, que tem um significado definido na linguagem de programação para → o programa de peça .

Parada exata

Quando uma parada exata é programada, a posição especificada no bloco é atingida de forma exata e, se necessário, muito lentamente. Para redução do tempo de aproximação são definidos limites de parada exata para avanço rápido ou → avanço.

Parada orientada de fuso

Parada do fuso em uma posição angular pré-determinada, por exemplo, para ser feita uma usinagem auxiliar em uma área específica.

Parâmetros R

Parâmetro de cálculo, que pode ser ajustado ou requisitado → no programa de usinagem para qualquer finalidade.

Peça

Peças a serem fornecidas ou usinadas pela máquina-ferramenta.

Peça bruta

Peça antes de ser usinada.

Ponto de referência

Ponto na máquina-ferramenta, que é referência para o sistema de medição dos → eixos da máquina.

Ponto fixo da máquina

Ponto único definido da máquina-ferramenta, por exemplo, ponto de referência da máquina.

Ponto zero da máquina

Ponto fixo da máquina-ferramenta, no qual permite atribuir à todo sistema de medição (derivado).

Ponto zero da peça

O ponto zero da peça forma o ponto de início para o → sistema de coordenadas da peça. É definido pela distância do → ponto zero da máquina.

Pré-coincidência

Troca de bloco ocorre quando a distância do trajeto aproxima-se de um valor que é igual à um delta especificado com relação à posição final.

Procura de blocos

Para testar um programa de usinagem ou no cancelamento do processo de usinagem, qualquer parte do programa pode ser selecionada utilizando a função "Procura de blocos", da qual o processo de usinagem pode iniciar ou continuar.

Programa de usinagem

Seqüência de instruções do comando NC, que em conjunto resultam na produção de uma → peça específica. E igualmente conduz a usinagem específica para a → peça bruta desejada.

Programa de usuário

Programa de usuário para o sistema de automação S7-300 são criados com a linguagem de programação STEP 7. O programa de usuário possui estrutura modular e consiste de blocos individuais.

Os tipos de básicos de blocos são:

- Blocos de códigos

  Estes blocos contêm as instruções em STEP 7.

- Blocos de dados

  Estes blocos contêm constantes e variáveis para o programa STEP 7.

Programa para transferência de dados PCIN

PCIN é um programa que auxilia no envio e recebimento de dados de usuário do CNC através de uma interface serial, como por exemplo programa de usinagem, corretor de ferramenta etc. O programa PCIN é executável em MS-DOS em computadores indústriais padrão.

Programa principal

Caracterizado com numeração ou indicação, outro programa principal, sub-rotina ou → ciclo podem ser chamados de dentro do → programa de usinagem.

Programação de CLP

O CLP é programado com o software **STEP 7**. O software de programação STEP 7 tem como base o sistema padrão **WINDOWS** e contém as funções de programação de STEP 5 com inovações.

Recuo orientado da ferramenta

RETTOOL: Com interrupções da usinagem (por exemplo: a quebra de ferramenta) a ferramenta pode ser recuada através de instruções de programação, em uma orientação definida pelo usuário através de uma distância definida.

Rede

Uma rede é a conexão de múltiplos S7-300 e outros terminais, por exemplo, uma PG, através → de cabos de conexão. Através da rede ocorre uma troca de dados entre os dispositivos conectados.

Reset geral

Através do reset geral toda a memória da → CPU é apagada:

- → Memória de trabalho
- Área de escrita/leitura da → memória de armazenamento

- → Memória de sistema
- → Memória de back-up

Retração rápida do contorno

Com uma interrupção o programa de usinagem do CNC pode-se introduzir um movimento, que torna possível uma retirada rápida da ferramenta de um contorno da peça que esteja sendo usinado. Adicionalmente pode-se parametrizar o ângulo e a distância de retração. Após uma retração rápida pode-se adicionalmente executar uma rotina de interrupção (SINUMERIK 840D).

Rosqueamento sem mandril de compensação

Com esta função permite-se fazer rosqueamento sem mandril de compensação. Através de métodos de interpolação do fuso como um eixo rotativo e o eixo de rosqueamento, o rosqueamento é feito com precisão na profundidade final da rosca. Por exemplo, roscas cegas (Condição: Fuso em operação de eixo).

Rotação

Componente de um → frame, que define uma rotação no sistema de coordenadas ao redor de um ângulo específico.

Rotina de interrupção

Rotinas de interrupção são → sub-rotinas especiais, que podem ser iniciadas através da execução de um evento (sinal externo) no processo de usinagem. Um bloco do programa de usinagem é interrompido, e a posição dos eixos são automaticamente armazenadas.

Saídas e entradas digitais rápidas

Em um entrada digital pode-se, por exemplo iniciar uma rotina rápida de CNC (rotina de interrupção). Através de uma saída digital do CNC pode-se rapidamente ativar funções de comutação controladas pelo programa (SINUMERIK 840D).

Sentença do programa de usinagem

Parte de um → programa de peça, demarcado através de Line Feed. Os → blocos principais e os → subblocos são diferenciados.

Sincronização

Instruções em → programas de usinagem para seqüências coordenadas em → canais diferentes em certos pontos de usinagem.

Sistema de coordenadas

Vide → Sistema de coordenadas de máquina, → Sistema de coordenada de peça

Sistema de coordenadas básico

Sistema de coordenadas cartesianas, que é gerado através de uma transformação do sistema de coordenadas da máquina.

O programador utiliza os nomes dos eixos do sistema de coordenadas base nos → programas de peça. O sistema de coordenadas base é paralelo ao → sistema de coordenadas de máquina caso nenhuma → transformação esteja ativa. A diferença para estes dois sistemas de coordenadas são os → identificadores dos eixos.

Sistema de coordenadas da máquina

Sistema de coordenadas, pelo qual os eixos da máquina-ferramenta são orientados.

Sistema de coordenadas da peça

O sistema de coordenadas de peça tem seu ponto de início no → ponto zero da peça. Para programação de usinagem no sistema de coordenadas de peça, a distância e direção da movimentação referem-se à este sistema.

Sistema de medição métrico

Sistema padrão de unidade: para comprimento, por exemplo, mm (milímetro), m (metro).

Sistema de unidade em polegadas

Sistema de unidade, as distâncias em "polegadas" e suas frações.

Softkey

Teclas, as quais são representadas por um campo na tela, e são dinamicamente adaptadas à situação atual de operação. As teclas (Softkeys) que estão disponíveis (livres) são atribuídas funções definidas pelo software.

SRT

Relação de transmissão

Subrotina

Seqüência de instruções de um → programa, que podem ser chamadas repetitivamente com o fornecimento de parâmetros diferentes. A chamada da sub-rotina é feita através do programa principal. Cada sub-rotina pode ser bloqueada para visualização e edição não autorizada. → Ciclos são uma forma de subrotina

Subrotina assíncrona

Programa de usinagem, que pode ser iniciado de forma assíncrona (independentemente) do programa atual, através de um sinal de interrupção (ex. um sinal "Entrada rápida de NC").

Tabela de compensação

Tabela de pontos de interpolação. É fornecido os valores de compensação dos eixos de compensação para posições selecionadas dos eixos base.

Técnica de macros

Agrupar uma certa quantidade de instruções sob um identificador. O identificador representa instruções agrupadas, em um programa.

Transformação

Deslocamento de ponto zero de um eixo absoluto ou incremental.

Unidade de offset de ferramenta ativa (TOA)

Cada → área de offset de ferramenta ativa pode conter várias unidades desta. A quantidade de TOA disponíveis é limitada de acordo com a quantidade de → canal ativos. Uma unidade TOA contém um bloco de dados de máquina e um bloco de dados de magazine. Adicionalmente também pode conter mais um bloco de dados de máquina de porta ferramenta (opcional).

Usinagem em superfície inclinada

Furação e fresamento na superfície da peça, que não está localizado no plano de coordenadas da máquina, pode ser conduzido confortavelmente com auxilio da função "usinagem em superfície inclinada".

Valor de compensação

Diferença entre a medição da posição do eixo medida, através do encoder, e a programada.

Variáveis de sistema

Uma variável que existe sem precisar da declaração do programador no → programa de usinagem. É definida através de um tipo de dado e de um nome de variável, precedida por **$**. Vide → Variáveis de usuário definidas.

Variáveis definidas pelo usuário

O usuário pode declarar suas próprias variáveis para qualquer propósito em → programas de usinagem ou bloco de dados (Dados de usuário globais). Uma definição contém o tipo de dados e os nomes das variáveis. Vide → Variáveis de sistema.

Velocidade de percurso

A máxima velocidade programável depende da resolução do campo de introdução. Uma resolução de por exemplo 0.1 mm permite um avanço programado de no máximo 1000 m/min.

Velocidade de transmissão

Velocidade para a transferência de dados (Bit/s).

WinSCP

WinSCP é um programa gratuito disponível para Windows, para transferência de arquivos.

Zona de Proteção

Área tri-dimensional dentro da → área de trabalho, na qual não é permitida a passagem da ponta da ferramenta.

Índice

"

"GET" automático, 121

$

$AA_ATOL, 484
$AA_COUP_ACT, 451, 492, 516
$AA_LEAD_SP, 515
$AA_LEAD_SV, 515
$AA_MOTEND, 268
$AA_TOFF[ ], 580
$AC_ACT_PROG_NET_TIME, 674
$AC_ACTUAL_PARTS, 677
$AC_BLOCKTYPE, 560
$AC_BLOCKTYPEINFO, 560
$AC_CTOL, 484
$AC_CUT_INV, 442
$AC_CUTMOD, 442
$AC_CUTMOD_ANG, 442
$AC_CUTTING_TIME, 674
$AC_CYCLE_TIME, 673
$AC_FIFO1, 558
$AC_MARKER, 553
$AC_OLD_PROG_NET_TIME, 674
$AC_OLD_PROG_NET_TIME_COUNT, 674
$AC_OPERATING_TIME, 673
$AC_OTOL, 484
$AC_PARAM, 554
$AC_PROG_NET_TIME_TRIGGER, 674
$AC_REQUIRED_PARTS, 677
$AC_SMAXVELO, 480
$AC_SMAXVELO_INFO, 480
$AC_SPECIAL_PARTS, 677
$AC_SPLITBLOCK, 560
$AC_STOLF, 487
$AC_TIMER, 557
$AC_TOTAL_PARTS, 677
$AN_POWERON_TIME, 673
$AN_SETUP_TIME, 673
$MC_COMPESS_VELO_TOL, 456
$P_AD, 442
$P_CTOL, 485
$P_CUT_INV, 442
$P_CUTMOD, 442
$P_CUTMOD_ANG, 442
$P_OTOL, 485
$P_STOLF, 487
$P_SUBPAR, 148
$P_TECCYCLE, 609
$PA_ATOL, 485
$R, 555
$Rn, 555
$SA_LEAD_TYPE, 515, 516
$SC_PA_ACTIV_IMMED, 214
$SN_PA_ACTIV_IMMED, 214
$TC_CARR1...14, 426
$TC_CARR18[m], 427, 431
$TC_DP1, 381
$TC_DP10, 382
$TC_DP11, 382
$TC_DP12, 382
$TC_DP13, 382
$TC_DP14, 382
$TC_DP15, 382
$TC_DP16, 382
$TC_DP17, 382
$TC_DP18, 382
$TC_DP19, 382
$TC_DP2, 381
$TC_DP20, 382
$TC_DP21, 382
$TC_DP22, 382
$TC_DP23, 382
$TC_DP24, 382
$TC_DP25, 382
$TC_DP3, 382
$TC_DP4, 382
$TC_DP5, 382
$TC_DP6, 382
$TC_DP7, 382
$TC_DP8, 382
$TC_DP9, 382
$TC_ECPxy, 386
$TC_SCPxy, 386
$TC_TPG1 ... 9, 647

***

* (função de cálculo), 59

/

/ (função de cálculo), 59

+

+ (função de cálculo), 59

<

< (operador de comparação), 62
<<, 68
<< (operador de encadeamento), 72
<= (operador de comparação), 62
<> (operador de comparação), 62

=

== (operador de comparação), 62

>

> (operador de comparação), 62
>= (operador de comparação), 62

A

A1, A2, 426
A2, 315
A3, 315
A4, 315, 322
A5, 315, 322
A6, 328
A7, 328
ABS, 59
Ação síncrona
    cancelar, 616
Ação sincronizada
    Ação, 545
    Área de validade, 541
    Condição, 543
    deletar, 616
    Elementos de comando, 540
    Posicionamento de eixo, 582
    Sintaxe, 540
ACC, 530
Ações sincronizadas
    Variáveis de pré-processamento, 548
    Variáveis de processamento principal, 548
    Vista geral das ações, 563
Acoplam, 445
Acoplamento axial de valor mestre, 511
Acoplamento de valor mestre
    a partir de ações sincronizadas estáticas, 512
    Acoplamento de valor real e valor nominal, 511, 514
    Sincronização do eixo mestre e eixo escravo, 514
Acoplamento de valor real, 527
Acoplamento de velocidade, 527
Acoplamento do valor nominal, 527
ACOS, 59
ACTBLOCNO, 160
ACTFRAME, 273
ADISPOSA, 266
Agrupamento de Spline, 237
Alarme, 679
    - Número, 679
    Comportamento, em ações síncronas,, 619
Alarmes de ciclo, 679
ALF, 111, 114
AND, 62
Ângulo de avanço, 316
Ângulo de rotação, 337
Ângulo de rotação 1, 2, 426
Ângulo de tangente de percurso, 603
Ângulo final, 337
Ângulo lateral, 316
Anulação de curso restante, 259, 567
Anulação de curso restante com preparação, 567
APR, 37
APRB, 37
Aprendizado de curvas características de compensação, 670
Aproximação do ponto de percurso mais próximo, 474
APRP, 37
APW, 37
APWB, 37
APWP, 37
Arquivo
    Informações, 137
Array, 43
Arredondamento, 142
AS, 194
ASIN, 59
A-Spline, 232
ASPLINE, 225
ASUP, 106
ATAN2, 59
ATOL, 482
Atributos de posição
    Programação indireta, 55
Atrito, 670

Atuais
Parâmetros, 147
Atuais Frames básicos de canal, 295
Atuais Frames básicos globais da NCU, 295
Atuais Frames de sistema, 294, 295
Atual 1º Frame básico no canal, 295
Atual Frame ajustável, 297
Atual Frame programável, 297
Atual Frame total, 297
AV, 524
Avaliação do tempo de uso, 604
Avanço
por eixo, 589
Avanço por eixo, 589
AX, 649
AXCTSWE, 657
AXCTSWED, 657
AXIS, 22
AXNAME, 71, 649
AXSTRING, 649
AXTOCHAN, 123
AXTOSPI, 649

B

B_AND, 62
B_NOT, 62
B_OR, 62
B_XOR, 62
B2, 315
B3, 315
B4, 315, 322
B5, 315, 322
B6, 328
B7, 328
BAUTO, 225
BFRAME, 273
Bifurcação
Programa, 83
BLOCK, 181
Bloco a bloco
- Supressão, 155
Bloco de parada, 466
Bloco de parâmetros servo
programável, 270
Bloqueio de leitura / entrada, 566
BLSYNC, 108
BNAT, 225
BOOL, 22
BOUND, 66
B-Spline, 233
BSPLINE, 225
BTAN, 225

C

C2, 315
C3, 315
C4, 315, 322
C5, 315, 322
C6, 328
C7, 328
CAC, 223
CACN, 223
CACP, 223
Caixa de transmissão eletrônica, 517
CALCDAT, 695
Cálculo de Frame
MEAFRAME, 289
CALL, 180
CALLPATH, 184, 201
Caminho de localização
Caminho de localização programável, 184
na chamada de subrotinas, 200
Para chamada de subrotina, 146
Campo
- Elemento, 43
CANCEL, 616
Caractere 0, 70
CASE, 83
CDC, 223
CFINE, 284
Chamada de subrotina com indicação de caminho e parâmetros, 184
Chamada modal de subrotina, 178
CHAN, 22
CHANDATA, 202
CHAR, 22
Chave fim de curso de SW, 590
CHECKSUM, 140
CHKDNO, 422
CIC, 223
Ciclos
Parametrizar ciclos de usuário, 189
Ciclos da SIEMENS, 679
Ciclos de tecnologia, 606
Concatenações, 612
Controlar execução cíclica ICYCOF, 611
em ações sincronizadas por blocos, 612
Estruturas de controle IF, 612
Instruções de salto GOTOP, GOTOF, GOTOB, 613
Parâmetros Default com valores iniciais, 610
Saltos incondicionais, 613
Cinemática

Resolvida, 431
cinemática resolvida, 427
CLEARM, 100, 599
CLRINT, 110
CMIRROR, 59, 278
COARSE, 524
COARSEA, 266
Código G
Programação indireta, 54
Coeficiente de polinômio, 243
Comandos de programação
Lista, 699
COMCAD, 239
COMPCAD, 346
COMPCURV, 239, 346
Compensação de erro de quadrante
Ativação do processo de aprendizado, 670
Desativação do processo de aprendizado, 670
Compensação do erro de quadrante
Readaptação, 671
COMPLETE, 202
COMPOF, 239, 346
COMPON, 239, 346, 456
Componente de Frame
FI, 280
MI, 280
SC, 280
TR, 280
Componente de Frame RT, 280
Compressor, 239
Compressor de blocos NC, 239
Condições gerais em transformações, 377
CONTDCON, 688
Contentor de eixo, 657
Contorno
- Codificação, 688
- Preparação, 682
- Tabela, 682, 688
Reaproximação, 468
CONTPRON, 682
controle
Estruturas de, 92
Controle AC, aditivo, 572
Controle AC, multiplicativo, 573
Controle de distância, 574
Controle de potência do laser, 570
Controle tangencial, 445
Coordenação de eixo, 590
Coordenação de programa
Nomes de canal, 102
Números de canal, 102
Correção de comprimento de ferramenta Online, 435, 580
Correção de ferramenta
Memória de correções, 381
Online, 397
Sistema de coordenadas para valores de desgaste, 392
Correção do raio da ferramenta
Desaceleração nos cantos, 265
Fresamento periférico 3D sem superfícies de limitação, 411
Correção do raio de ferramenta 3D, 402
3D Ponto de intersecção das eqüidistantes, 410
Cantos internos/externos, 409
Círculo de transição, 410
Fresamento de topo, 405
Fresamento periférico, 404
Correções de ferramenta 3D, 406
Correção na trajetória, 407
Curvatura de trajetória, 408
Fresamento periférico:com superfícies de limitação, 411
Orientação da ferramenta, 416
Procedimento de ponto de intersecção, 410
Profundidade de imersão, 408
Corretores de ferramenta
Online, 577
COS, 59
COUPDEF, 524
COUPDEL, 524
COUPOF, 524
COUPOFS, 524
COUPON, 524
COUPONC, 524
COUPRES, 524
CP, 369
CPROT, 211
CPROTDEF, 207
Critério de fim de movimento programável, 266
CROT, 59, 278
CSCALE, 59, 278
C-Spline, 234
CSPLINE, 225
CT, 657
CTAB, 504
CTABDEF, 493
CTABDEL, 500
CTABEND, 493
CTABEXISTS, 500
CTABFNO, 509
CTABFPOL, 509

CTABFSEG, 509
CTABID, 503
CTABINV, 504
CTABISLOCK, 503
CTABLOCK, 501
CTABMEMTYP, 503
CTABMPOL, 509
CTABMSEG, 509
CTABNO, 509
CTABNOMEM, 509
CTABPERIOD, 503
CTABPOL, 509
CTABPOLID, 509
CTABSEG, 509
CTABSEGID, 509
CTABSEV, 504
CTABSSV, 504
CTABTEP, 504
CTABTEV, 504
CTABTMAX, 504
CTABTMIN, 504
CTABTSP, 504
CTABTSV, 504
CTABUNLOCK, 501
CTOL, 482
CTRANS, 59, 278, 284
CUT3DC, 402, 407
CUT3DCC, 411
CUT3DCCD, 411
CUT3DF, 402
CUT3DFF, 402
CUT3DFS, 402
CUTMOD, 438

D

Dados de círculo
    calcular, 695
de estruturas de controle
    Comportamento do tempo de processamento, 93
DEF, 22, 43, 606
DEFAULT, 83
DEFINE, 606
DEFINE ... AS, 194
Definição de campo, 43
Definição de polinômio, 569
Definir valor real, 591
DELAYFSTOF, 460
DELAYFSTON, 460
DELDL, 387
DELDTG, 567
DELETE, 129
Desaceleração de cantos em cantos internos, 265
Desaceleração de cantos em todos os cantos, 265
Desbaste, 681
Deslocamento aproximado, 284
Deslocamento de ponto zero
    Deslocamento externo de ponto zero, 286
    PRESETON, 287
Deslocamento de Preset, 287
Deslocamento externo de ponto zero, 286
Deslocamento fino, 284
Deslocamento PTP cartesiano, 306
Detectar e localizar áreas que não devem ser pesquisadas, 466
DISABLE, 109
DISPLOF, 160
DISPLON, 160
Disponibilidade
    Dependente do sistema, 5
Disponibilidade dependente de
    Sistema, 5
DISPR, 468
DIV, 59
DL, 385
DO, 545
DV, 524

E

EAUTO, 225
EG
    Caixa de transmissão eletrônica, 517
EGDEF, 517
EGDEL, 523
EGOFC, 522
EGOFS, 522
EGON, 518
EGONSYN, 518
EGONSYNE, 518
Eixo
    - Troca, 118
    aceitar diretamente, 118
    De fixação, 657
    Inclinação (TRAANG), 364
    local, 658
    Movimento acoplado, 491
Eixo escravo, 511
Eixo geométrico
    comutar, 652
Eixo inclinado, TRAANG, 305
Eixo lincado, 658
Eixo mestre, 511
EixoE, 445, 511

EixoM, 445, 511
Eixos de comando, 581
Eixos de orientação, 315, 322, 325
Eixos de rotação
  Vetores de direção V1, V2, 426
  Vetores de distância l1, l2, 426
Eixos FGROUP, 248
Eixos geométricos comutáveis, 652
Elemento de contorno
  executar, 694
ELSE, 93
ENABLE, 109
ENAT, 225
Encadeamento
  de Strings, 72
Encosto fixo, 600
Endereços
  Programação indireta, 51
Endereços OEM, 264
ENDFOR, 95
ENDIF, 93
ENDLABEL, 85
ENDLOOP, 95
ENDPROC, 576
ENDWHILE, 97
Estado do acoplamento, 492, 516
Estado do apalpador de medição, 261
Estado do pedido de medição, 261
Estampagem, 635, 640
ETAN, 225
Evento de disparo
  Durante a medição, 258
EVERY, 543
EXECSTRING, 57
EXECTAB, 694
EXECUTE, 207, 697
Exibição de bloco, 183
  suprimir, 160
EXP, 59
EXTCALL, 186
EXTERN, 175

F

F10, 207
F3, 670
FA, 524, 589
FALSE, 22
Fator de acoplamento, 489
Fator de tolerância G0, 486
FCTDEF, 397, 569
FCUB, 452
FENDNORM, 265
Ferramenta
  - Controle, específico de retificação, 647
  - Correção do raio, 389
  - Correções, aditivas, 385
  - Memória de correções, 381
  - Orientação na mudança de Frames, 434
  - Parâmetros, 381
  Correção do comprimento, 432
FIFOCTRL, 457
FILEDATE, 137
FILEINFO, 137
FILESIZE, 137
FILESTAT, 137
FILETIME, 137
FINE, 524
FINEA, 266
FLIN, 452
FNORM, 452
FOCOF, 600
FOCON, 600
Folga, 670
FOR, 95
Formal
  Parâmetro, 146
Formas de fresa, 406
FPO, 452
FPR, 523
Frame
  chamar, 281
  Encadeamento de Frames, 299
FRAME, 22
Frame básico total, 296
Frames
  atribuir, 282
  Encadeamento de Frames, 283
Frames ajustáveis globais da NCU, 293
Frames básicos globais da NCU, 293
Frames específicos de canal, 294
Fresa
  - Ponta (FS), 409
  - Ponto auxiliar (FH), 409
Fresamento de topo, 402, 405
Fresamento de topo 3D, 321
  Curvatura de trajetória através de vetores normais de superfície, 322
Fresamento periférico, 403, 404
Fresamento periférico (3D)
  com superfícies de limitação, 411
Fresamento periférico 3D com superfícies de limitação, 411
FROM, 543

FTOC, 577
FTOCOF, 397
FTOCON, 397
Função de avaliação, 571
Função de medição ampliada, 369
Funções auxiliares, 565, 640
Funções OEM, 264
Fuso
- Troca, 118
Fuso sincronizado, 524
- Par, 524
Definição de par, 530
Relação de transmissão kÜ, 530
FXS, 600
FXST, 600
FXSW, 600

G

G05, 367
G07, 367
G40, 402
G450, 409
G451, 409
G62, 265
G621, 265
GEOAX, 652
GET, 118
GETACTTD, 424
GETD, 118
GETDNO, 423
GOTO, 80
GOTOB, 80
GOTOC, 80
GOTOF, 80
GOTOS, 79
GP, 55
Grupo de movimento acoplado, 489
GUD, 22, 198

I

I1,I2, 426
ICYCOF, 611
ICYCON, 611
ID, 541
Identificador de eixo default, 552
IDS, 541
IF, 80, 93
IFRAME, 273
II1,II2, 630
INDEX, 75
Indicação de caminho
Absoluto, 100
relativo, 100
Indicador de interrupção automático, 467
Índice de campo, 45
INICF, 22
Inicialização
De campos, 43
de variáveis de campo, 598
Inicialização do curso, 638
Iniciar/parar eixo, 585
INIPO, 22
INIRE, 22
INIT, 100
INITIAL, 202
INITIAL_INI, 202
Instrução de salto
CASE, 83
Instruções
Lista, 699
INT, 22
Interpolação de orientação, 344
Interpolação de polinômios, 242
Polinômio de denominador, 246
Interpolação do vetor de rotação, 336, 342
INTERSEC, 692
IPOBRKA, 266
IPOENDA, 266
IPOSTOP, 524
IPTRLOCK, 465
IPTRUNLOCK, 465
ISAXIS, 649
ISD, 402, 407
ISFILE, 134
ISNUMBER, 71
ISOCALL, 183
ISVAR, 667

J

JERKLIM, 477

L

L..., 172
Label, 85
LEAD, 315
LEADOF, 511
LIFTFAST, 111
LLI, 34

LLIMIT, 569
LN, 59
LOCK, 614
LOOP, 95
Loop contador, 95
Loop de programa
  Loop contador, 95
  Loop final, 95
  Loop IF, 93
  Loop REPEAT, 98
  Loop WHILE, 97
Loop infinito, 95
LUD, 22

M

M, 428
M17, 164
M30, 164
Macro, 194
Marcadores de espera, 599
MASLDEF, 535
MASLDEL, 535
MASLOF, 535
MASLOFS, 535
MASLON, 535
MATCH, 75
MAXVAL, 66
MCALL, 178
MD20800, 164
MD37400, 451
MEAC, 254
MEAFRAME, 289, 293
MEAS, 251
MEASA, 254
MEAW, 251
MEAWA, 254
Medição, 596
Memória
  de trabalho, 202
  Memória de programa, 197
Memória de correções, 381
Memória de programa, 197
  Diretórios padrão, 198
  Tipos de arquivo, 198
Memória de trabalho, 202
  Áreas de dados, 202
MINDEX, 75
MINVAL, 66
MIRROR, 273
MMC, 672
MOD, 59
MODAXVAL, 649
Modo de operação
  Durante a medição, 259
MOV, 585
Movimento acoplado, 489, 593
  Limitação de dinâmica, 492
Movimento alternado
  Área de reversão, 630
  Penetração no ponto de reversão, 632
  Ponto de reversão, 630
  Suprimir a penetração, 630
Movimento de eixo individual, 645
Movimentos de fuso, 592
Movimentos de posicionamento, 581
MPF, 198, 670
MU, 366
Mudança de estação/posição, 657
MZ, 366

N

na chamada de subrotina
  Transferência de parâmetro, 147
NCK, 22
NEWCONF, 125
Nível de aninhamento
  de estruturas de controle, 92
NOC, 524
NOT, 62
NPROT, 211
NPROTDEF, 207
NUMBER, 71
Número D
  atribuir livremente, 422
Número de corte, 422
Número de identificação, 541
Números D
  renomear, 423
  verificar, 422
NUT=ângulo, 328

O

OEMIPO1/2, 264
OFFN, 352, 355
Offset angular/incremento angular dos eixos de rotação, 429
Offset dos eixos de rotação, 429
Offset normal de contorno OFFN, 361
Operadores de comparação, 62
Operadores lógicos, 62

OR, 62
ORIAXES, 325, 342
ORIC, 416
ORICONCCW, 328, 342
ORICONCW, 328, 342
ORICONIO, 328, 342
ORICONTO, 328, 342
ORICURVE, 331, 342
ORID, 416
Orientação
  Eixos, 328
  Interpolação, 329
Orientação da ferramenta, 416
Orientação relativa à trajetória
  Inserção de blocos intermediários, 345
  Rotação da orientação da ferramenta, 341
  Rotação do vetor de orientação, 342
  Rotações da ferramenta, 340
ORIEULER, 325, 342
ORIMKS, 322, 325
ORIPATH, 340
ORIPATHS, 340, 344
ORIPLANE, 328, 342
ORIRESET(A, B, C), 313
ORIROTA, 336
ORIROTC, 336, 342
ORIROTR, 336
ORIROTT, 336
ORIRPY, 325, 342
ORIRPY2, 325
ORIS, 416
ORISOF, 350
ORISON, 350
ORIVECT, 325, 342
ORIVIRT1, 325, 342
ORIVIRT2, 325, 342
ORIWKS, 322, 325
OS, 621
OSB, 621
OSC, 416
Oscilação
  Assíncrono(a), 621
  controlar através de ação sincronizada, 627
  Oscilação assíncrona, 621
  Oscilação sincronizada, 627
  Penetração parcial, 630
Oscilação assíncrona, 621
Oscilação sincronizada
  Ações sincronizadas, 631
  Atribuição do eixo oscilante e eixo de penetração, 630
  Avaliação, ciclo de interpolação, 633
  Definir penetrações, 630
  Movimento de penetração, 631
  Parada no ponto de reversão, 632
  Penetração na área de reversão, 631
  Próxima penetração parcial, 633
OSCILL, 627, 630
OSCTRL, 621
OSD, 416
OSE, 621
OSNSC, 621
OSOF, 416
OSP1, 621
OSP2, 621
OSS, 416
OSSE, 416
OST, 416
OST1, 621
OST2, 621
OTOL, 482
Override
  atual, 603
  resultante, 603
OVRA, 530

P

P..., 177
Parada de pré-processamento, 567
Parâmetro
  Transferência na chamada de subrotina, 175
Parâmetro Call-by-Value
  para ciclos de tecnologia, 610
Parâmetro de ação sincronizada, 554
Parâmetros
  Ferramenta, 381
Parâmetros de cálculo
  Número n, 18, 20
Parâmetros de cálculo (R), 18, 20
Parâmetros R, 555
parte do programa
  Repetição de, 85
PCALL, 184
PDELAYOF, 635
PDELAYON, 635
Peça
  Contador de, 677
Peça de trabalho
  - Diretório principal, 198
  - Diretórios, 198
Penetração
  - Movimento, 632
  Eixo, 628

PFRAME, 273
PHI, 328, 335
PHU, 35
PL, 225, 242
PO, 242
PO[PHI], 335, 340
PO[PHI]=(a2, a3, a4, a5), 328
PO[PSI], 335, 340
PO[PSI]=(b2, b3, b4, b5), 328
PO[THT], 335, 340
PO[XH], 335
PO[XH]=(xe, x2, x3, x4, x5), 331
PO[YH], 335
PO[YH]=(ye, y2, y3, y4, y5), 331
PO[ZH], 335
PO[ZH]=(ze, z2, z3, z4, z5), 331
Polinômio de denominador, 246
POLY, 242
POLYPATH, 242
PON, 644
PONS, 635
Porta-ferramenta, 432
- Cinemática, 426
Deletar/alterar/ler dados, 431
Orientável, 432
Porta-ferramenta orientável, 426
Número do porta-ferramenta, 428
Variáveis de sistema, 427
POS, 582
POSFS, 524
Posição básica da orientação da ferramenta
ORIRESET, 314
Posição de desativação, 533, 534
Posição mínima/máxima do eixo de rotação, 429
Posicionamento de eixo
Posição de referência especificada, 584
Posições singulares, 324
POSP, 627
POSRANGE, 584
POT, 59
Preparação de contorno
Resposta de erro, 697
PREPRO, 163
Pré-processamento
Memória de, 457
Memória de, 457
PRESETON, 287, 591
Primeiro Frame básico no canal, 294
PRIO, 108, 111
PRLOC, 22
PROC, 150
Profundidade de imersão, 408
Profundidade de imersão (ISD), 402
Programa
- Memória, 198
- Saltos, 80
de inicialização, 202
Repetição de, 177
Tempos de processamento, 673
Programa de inicialização, 202
Programação da orientação, 326, 343
Programação indireta, 55
de códigos G, 54
de endereços, 51
Programar eixo inclinado
G05, G07, 367
Programar rotações do vetor de orientação através do THETA, 336
Proteção
Áreas, 207
PSI, 328, 335
PTP, 369, 373
PTP com TRANSMIT, 373
PTPG0, 373
PUD, 22
PUNCHACC, 635
Puncionamento, 635, 640
PUTFTOC, 397
PUTFTOCF, 397
PW, 225

Q

QECDAT, 670
QECLRN, 670
QECLRNOF, 670
QECLRNON, 670
QECTEST, 670
QFK, 670

R

R..., 18, 20
RDISABLE, 566
READ, 131
REAL, 22
Reaproximação no contorno
Aproximação com nova ferramenta, 474
Ponto de reaproximação, 472
REDEF, 28
Referência angular, 531
Referência do percurso
Ajustável, 248

Refpos, 584
RELEASE, 118
REP, 43, 598
REPEAT, 85, 98
REPEATB, 85
Repetição de partes do programa
com programação indireta CALL, 181
REPOS, 106
REPOSA, 468
REPOSH, 468
REPOSHA, 468
REPOSL, 468
REPOSQ, 468
REPOSQA, 468
RESET, 614
RET, 165, 166
Retração rápida do contorno, 111
Reversão
Ponto, 627
RINDEX, 75
RMB, 468
RME, 468
RMI, 468
RMN, 468
Rotina de interrupção, 106
Ativação / desativação, 109
Atribuir e iniciar, 108
Deletar, 110
Movimento de retrocesso, 114
Nova atribuição, 109
Retração rápida do contorno, 111
Salvar funções G modais, 107
Sentido de deslocamento programável, 112, 114
Rotinas de conversão, 549
ROUND, 59
ROUNDUP, 142

S

S1, S2, 524
Salto
- Condição, 81
- Destino, 80
- Instrução, 80
- Marcador, 81
Marcadores de, 85
Vai para o início do programa, 79
SAVE, 154
SBLOF, 155
SBLON, 155
SC, 445
SCPARA, 270
SD, 225
SD42475, 347
SD42476, 347
SD42477, 347
SD42678, 350
SD42680, 350
SD42900, 391
SD42910, 391
SD42920, 391
SD42930, 392
SD42935, 394
SD42940, 395, 441
SD42984, 439
SEFORM, 205
Segmentação automática de curso, 640
Segmentação de curso, 644
Segmentação de curso para eixos de percurso, 643
Selecionar caracteres individuais, 77
SET, 43, 598
SETAL, 599, 679
SETDNO, 423
SETINT, 108
SETM, 100, 599
Simulação de valor mestre, 515
SIN, 59
Sincronismo de posição, 525
Sincronização
(fino), 527
aproximado, 527
Solavanco
- Correção, 477
SON, 635, 643, 644
SONS, 635
SPATH, 248
SPF, 198, 670
SPI, 649
SPIF1, 635
SPIF2, 635
Spline
- Interpolação, 225
- Tipos, 231
SPLINEPATH, 237
SPN, 640
SPOF, 635
SPOS, 524
SPP, 640
SQRT, 59
START, 100
STARTFIFO, 457
STAT, 369, 373
STOLF, 486
STOPFIFO, 457

STOPRE, 457
STOPREOF, 567
String
- Comprimento, 75
- Encadeamento, 72
- Operações, 70
STRING, 22
STRINGIS, 663
STRINGVAR, 77
STRLEN, 75
suavização
Orientação de ferramenta,, 350
Suavização
Do decurso de orientação, 350
Suavização do decurso de orientação, 341, 345
Subrotina, 143
- Chamada com transferência de parâmetros, 175
- Chamada sem transferência de parâmetros, 172
- Chamada, indireta, 180
- Chamada, modal, 178
- Repetição, 177
Caminho de localização programável, 184
Nome da, 144
Salto de retorno, parametrizável, 166
SUBSTR, 77
SYNFCT, 571
SYNR, 22
SYNRW, 22
SYNW, 22

T

TAN, 59
TANG, 445
TANGDEL, 445
TANGOF, 445
TANGON, 445
TCARR, 432
TCOABS, 432
TCOFR, 432
TCOFRX, 432
TCOFRY, 432
TCOFRZ, 432
Tempo de processamento, 674
Tempo necessário
Ações sincronizadas, 604
Tempo restante
para uma peça de trabalho, 675
THETA, 335, 336
TILT, 315
Tipo de acoplamento, 527
Tipo de cinemática, 431
Tipo de cinemática M, 431
Tipo de cinemática P, 431
Tipo de cinemática T, 431
Tipos de transformação
Função geral, 301
TLIFT, 445
TMOF, 647
TMON, 647
TOFFOF, 435
TOFFOF, 580
TOFFON, 435
TOFFON, 580
Tolerância
com G0, 486
TOLOWER, 74
Torção, 670
TOUPPER, 74
TOWBCS, 392
TOWKCS, 392
TOWMCS, 392
TOWSTD, 392
TOWTCS, 392
TOWWCS, 392
TRAANG, 364
TRACON, 379
TRACYL, 355, 361
TRAFOOF, 378
TRAILOF, 489
TRAILON, 489
Transformação
Eixo inclinado, 364
Transformação cinemática TRANSMIT, TRACYL e TRAANG, 305
Transformação com eixo linear giratório, 311
Transformação de curvas da superfície periférica do cilindro, 355, 356
Offset normal de contorno OFFN, 361
Transformação de orientação TRAORI
Cinemática da máquina, 304
Movimentos de deslocamento e movimentos de orientação, 303
Programação da orientação, 313
Transformações genéricas de 5 e 6 eixos, 304
Variantes da programação da orientação, 314
Transformação de superfície cilíndrica, 305
Transformação polar, 305
Transformação TRACYL, 357
Transformação TRANSMIT, 353
Transformação, 5 eixos
Programação da orientação da ferramenta com LEAD e TILT, 320
Programação do vetor de direção, 319

Transformação, de cinco eixos
Programação através de LEAD/TILT, 315
Programação da curvatura de trajetória em vetores normais de superfície, 321
Programação em ângulos eulerianos, 318
Programação em ângulos RPY, 319
Transformações
encadeadas, 379
Posição básica da orientação da ferramenta independente da cinemática, 302
Transformação de orientação, 301
Transformação de três, quatro e cinco eixos TRAORI, 302
Transformações cinemáticas, 302
Transformações de três e quatro eixos, 312
Transformações encadeadas, 303
TRANSMIT, 352, 355, 373
TRAORI, 309, 312
Trecho, 640
Trechos, 640
Troca de eixos, 123
Aceitar eixo, 121
Ajustar o comportamento de forma modificável, 122
Liberar eixo, 121
Requisitos, 121
sem parada de pré-processamento, 122
sem sincronização, 120
solicitar e liberar através de ações sincronizadas, 586
TRUE, 22
TRUNC, 59
TU, 369, 373

U

U1,U2, 630
uc.com, ciclos de usuário, 190
ULI, 34
ULIMIT, 569
UNLOCK, 614
UNTIL, 98
UPATH, 248

V

V1,V2, 426
Valor de ajuste, 386
Valor de desgaste, 386
Valor mestre
Acoplamento, 594
VAR, 151
Variáveis
Conversão de tipos, 70
Definição, 22
definidas pelo usuário, 22
Nome, 24
Nome de, 28
Tipo de, 22
Variáveis de sistema, 548
Variáveis GUD
Apto para ações síncronas, 551
Variável
Conversão de tipos, 69
Variável de marcador, 553
Variável de temporizador, 557
Variável FIFO, 558
Variável Frame, 271
Atribuição com os comandos G54 até G599, 277
Atribuir valores, 278
Chamada de transformações de coordenadas, 271
Definição de novos Frames, 283
Deslocamentos de ponto zero G54 até G599, 277
Variável Frame pré-definida, 273, 281
VELOLIM, 478
Vista geral
Frames ativos no canal, 295

W

WAIT, 100
WAITC, 524
WAITE, 100
WAITENC, 661
WAITM, 100
WAITMC, 100
WHEN, 543
WHEN-DO, 631
WHENEVER, 543
WHENEVER-DO, 631
WHILE, 97
Winlimit, 584
WRITE, 126

X

xe, ye, ze, 331
XH YH ZH, 331
xi, yi, zi, 331
XOR, 62

A

α, 364